

Semestrální práce

Zřízení vnitřního informačního systému pro Karate Klub

Grigoriants Karen (@grigoka2)

Voropaeva Alina (@voropali)

Luzan Anna (@luzanann)

Obsah

Obsah.....	1
Popis aplikace.....	3
Motivace.....	3
Strategický záměr (stav „TO BE“).....	3
Stav „AS IS“	4
SWOT.....	4
Analýza 5F.....	4
Analýza PEST(E).....	5
Funkční požadavky.....	6
Nefunkční požadavky.....	6
Seznam uživatelů.....	6
Případy užití.....	7
Diagram nasazení.....	0
Diagram komponent.....	0
Rozbor a výběr alternativ návrhu řešení.....	0
WBS – rozdělení projektu na dílčí procesy.....	0
Zdroje.....	0
Normy a standardy.....	0
Matice zodpovědnosti.....	0
Harmonogram GANTT.....	0
Analýza rizik FMEA.....	0
Znovupoužitelnost.....	0
Metriky.....	0

Plán odbavení.....	0
Plán podpory.....	0
Vyhodnocení.....	2

Popis aplikace

Aplikace pro správu karate klubu je navržena jako komplexní řešení pro efektivní správu a organizaci karate klubu. Jedná se o samostatný softwarový produkt, který má za cíl usnadnit a vylepšit správu členů klubu, jejich úspěchů a finančních transakcí. Systém interaguje s databází pro ukládání a získávání informací týkajících se členů, plateb, turnajů a oznámení. Klíčové komponenty aplikace zahrnují správu členů, modul turnajů a systém oznámení. Uživatelé mohou využívat funkce jako registrace členů, správa turnajů, posílání oznámení, hledání a filtrování informací, sledování bodů a úspěchů členů a realizaci rolí s rozdílným přístupem na základě jejich úrovně v klubu. Aplikace nabízí univerzální použitelnost pro různé karate kluby a je vizualizována pomocí UML diagramu tříd, který zobrazuje organizaci a interakci hlavních komponent systému.

Motivace

Motivací pro vytvoření aplikace pro správu karate klubu je poskytnout komplexní a efektivní řešení pro administraci a organizaci klubu, které usnadní a zlepší každodenní procesy spojené s členstvím, úspěchy a komunikací. Tato aplikace má za cíl eliminovat manuální procesy a zjednodušit správu členů, registraci turnajů a šíření důležitých oznámení. Vytvoření aplikace také umožňuje zvýšit transparentnost a přehlednost informací pro členy klubu a usnadnit organizaci a průběh karate turnajů. Díky role-based přístupu poskytuje aplikace zabezpečené prostředí pro správu dat a umožňuje personalizované používání funkcí pro různé uživatelské profily. Celkově je motivací pro vytvoření aplikace pro správu karate klubu zlepšení efektivity, transparentnosti a komunikace v rámci karate komunity.

Strategický záměr (stav "TO BE"):

Cílem strategického záměru je přetvořit aplikaci pro správu karate klubu do stavu "TO BE", který bude reflektovat optimální stav po implementaci strategických změn a vylepšení.

Rozšířená funkcionální:

Implementace rozšířených funkcí pro lepší správu a organizaci klubu, včetně možnosti plánování tréninků, sledování pokroku členů, a integrace platebních systémů pro snadnější správu financí.

Vylepšené uživatelské rozhraní:

Redesign uživatelského rozhraní s důrazem na intuitivnost a jednoduchost použití, aby bylo pro uživatele snazší a příjemnější pracovat s aplikací.

Zvýšení personalizace:

Poskytnutí možnosti personalizace uživatelských zkušeností, včetně možnosti nastavení preferencí, upravování profilů a možnosti přizpůsobení aplikace potřebám konkrétního klubu.

Optimalizace výkonu a bezpečnosti:

Optimalizace výkonu aplikace pro zlepšení rychlosti a spolehlivosti, a zároveň zajištění bezpečnosti dat prostřednictvím zabezpečených přístupových kontrol a šifrování dat.

Partnerství a expanze:

Navázání partnerství s dalšími karate kluby, organizacemi a trenéry pro sdílení zkušeností a vytváření síťových efektů, a zároveň rozšíření dostupnosti aplikace na mezinárodní úroveň.

Cílem těchto strategických změn je dosáhnout optimalizovaného stavu aplikace, který bude lépe vyhovovat potřebám uživatelů a přinese větší hodnotu pro karate komunitu jako celek

Stav "AS IS":

Lepší správa členů: V současné době karate klub využívá ruční metody správy členů, jako jsou papírové seznamy a tabulky. Přejít na systém správy členů umožní klubu efektivněji uchovávat informace o svých členech, včetně jejich osobních údajů, členského statusu a dosažených úrovní.

Organizace turnajů: Systém umožní klubu snadno organizovat a sledovat průběh karate turnajů. To zlepší celkovou organizaci a efektivitu turnajů a poskytne lepší zážitek pro účastníky.

Transparentnost úspěchů členů: Karate klub bude mít možnost lépe sledovat a spravovat úspěchy svých členů, včetně získaných bodů a dosažených úrovní. To může poskytnout motivaci pro členy klubu a zlepšit jejich zapojení do tréninků a turnajů.

Univerzalita a centralizace dat: Přejít na systém správy karate klubu umožní centralizaci dat, což usnadní přístup k důležitým informacím o členech, turnajích a událostech. To může vytvořit jednotnou a konzistentní databázi, která bude snadno dostupná pro všechny administrátory klubu.

Zvýšení efektivity a produktivity: Implementace systému správy karate klubu povede ke zvýšení efektivity a produktivity administrativních procesů. Méně času bude stráveno ručním zpracováním dat a administrátorským úkolem, což umožní zaměřit se více na podporu členů klubu a rozvoj karate komunity

SWOT

Strengths	Weaknesses
<ul style="list-style-type: none"> Centralizovaná správa dat Možnost automatizace procesů Podpora efektivní komunikace 	<ul style="list-style-type: none"> Nepřítomnost systému v současnosti Potřeba školení personálu Omezené finanční zdroje pro implementaci
Opportunities	Threats
<ul style="list-style-type: none"> Zlepšení organizace klubu Možnost rozšíření funkcí Partnerství s dalšími kluby pro sdílení zkušeností 	<ul style="list-style-type: none"> Technické problémy při implementaci Potenciální odpor uživatelů k novému systému Konkurence od jiných softwarových řešení

Analýza 5F

Konkurence

Konkurenčních firem je velké množství. Mezi hlavní konkurenty patří již zavedené aplikace pro správu sportovních klubů, které již mají svůj podíl na trhu a nabízejí podobné funkce a služby.

Síly dodavatelů

Pro společnost je výhodou možnost volby dodavatelů pro potřebné technologické a informační prvky pro vývoj aplikace. Tato nezávislost a flexibilita umožňuje společnosti efektivně reagovat na změny v potřebách trhu a zákazníků.

Síly odběratelů

Primární cílovou skupinou jsou karate kluby a organizace. Je důležité sledovat jejich potřeby a preference při vývoji aplikace a zajistit, aby aplikace splňovala jejich očekávání. Zlepšení uživatelského rozhraní a poskytnutí personalizovaných funkcí může přispět k získání a udržení loajality uživatelů.

Substituté

Možnost, že karate kluby mohou používat jiné softwarové nástroje nebo opensource řešení místo aplikace nabízené společností.

Nově příchozí

Nové firmy nebo konkurenční produkty mohou vstoupit na trh s inovativními funkcemi nebo nižšími cenami, což může ohrozit pozici společnosti na trhu.

Analýza Pest(E)

Politické

Situace stabilní.

Politická situace nemá významný dopad na provoz aplikace

Ekonomické

Situace stabilní v oboru IT.

Tržní potenciál je stabilní a umožňuje společnosti plánovat dlouhodobě.

Společenské

Situace stabilní.

Aplikace není citlivá na sociální změny.

Technologické

Situace stabilní.

Technologické prostředí umožňuje využití moderních technologií při vývoji aplikace a zajištění bezpečnosti a výkonu.

Ekologické

IT obor softwarových aplikací není závislý na přírodních zdrojích a má minimální dopad na životní prostředí. Aplikace pro správu karate klubu nepředstavuje žádné ekologické riziko nebo dopad.

Funkční požadavky:

- Registrace členů
- Správa turnajů
- Systém oznámení
- Hledání a filtrování informací
- Sledování bodů a úspěchů členů
- Realizace rolí s rozdílným přístupem
- Správa dokumentací
- Upload dokumentu

Nefunkční požadavky:

- Aplikace musí mít webové rozhraní optimalizované v prohlížečích Firefox, Chrome, Opera
- Optimalizace výkonu aplikace
- Zajištění bezpečnosti dat
- "User-friendly" GUI
- Možnost rozšíření aplikace v budoucnu
- Centrální úložiště dat (MySQL)

Seznam uživatelů:

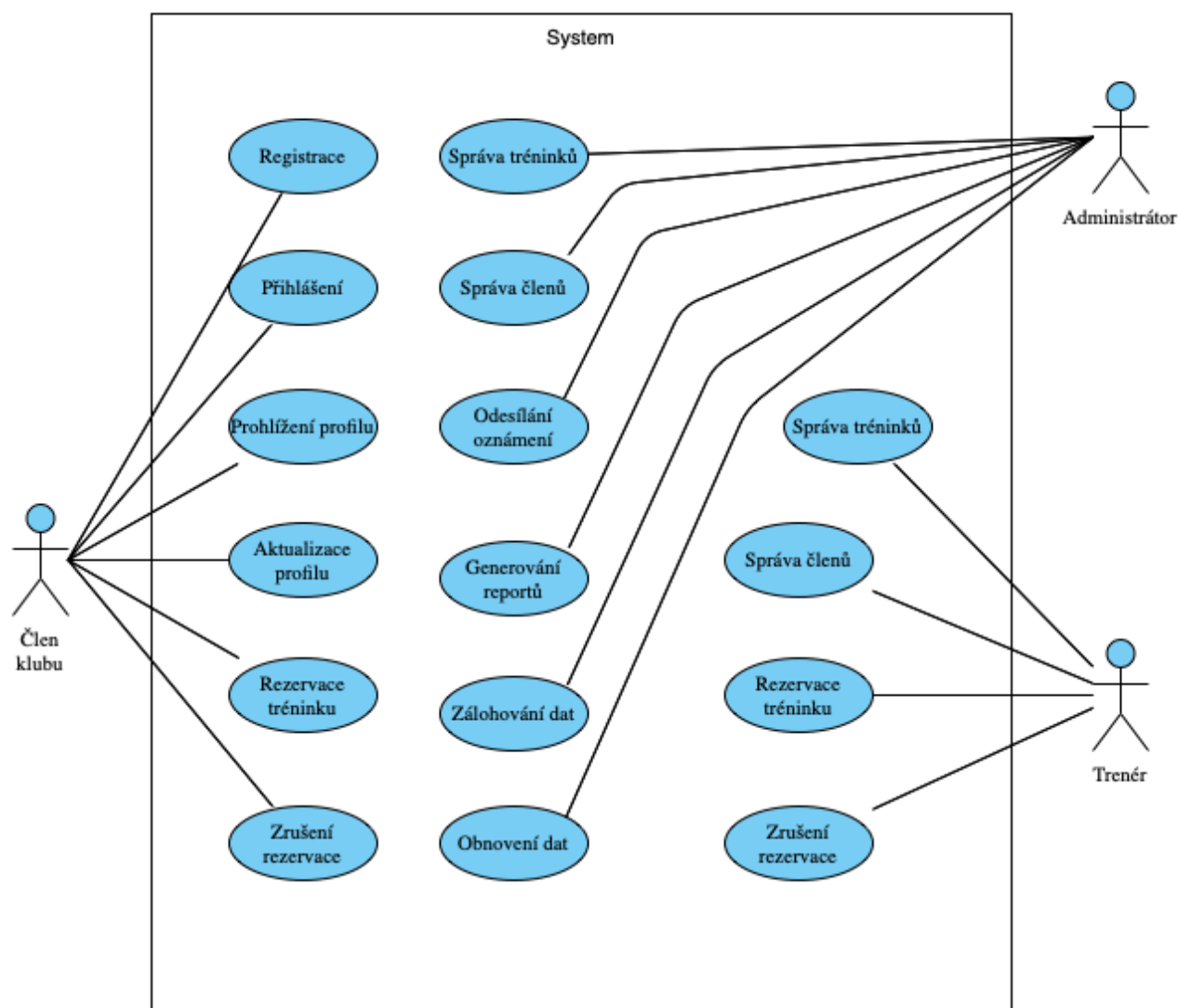
Systém podporuje 3 typy uživatelů:

Host: Může se zaregistrovat a přihlásit do systému.

Uživatel: Může spravovat své členství, účastnit se turnajů a sledovat své úspěchy.

Administrátor: Má plná práva pro správu všech funkcí systému.

Případy užití:



Registrace nového člena

- Popis: Tento případ užití pokrývá proces, kterým nový člen klubu prochází při registraci do systému.
- Kroky:
 1. Nový člen přistoupí k registrační stránce na webovém rozhraní.
 2. Vyplní požadované osobní údaje (jméno, příjmení, datum narození, kontaktní informace atd.).
 3. Ověří správnost zadaných údajů a odešle registrační formulář.
 4. Systém zkontroluje údaje a vytvoří nový členský účet.
 5. Nový člen obdrží potvrzovací e-mail s přihlašovacími údaji.

Plánování tréninku

- Popis: Tento případ užití popisuje proces, kterým trenér naplánuje nový trénink v systému.
- Kroky:

1. Trenér se přihlásí do systému jako administrátor.
2. Vybere modul pro plánování tréninků.
3. Zadá datum, čas, místo a další detaily tréninku.
4. Zvolí cílovou skupinu členů, kteří se tréninku zúčastní.
5. Uloží plánovaný trénink do systému.
6. Systém automaticky rozešle oznámení vybraným členům o novém tréninku.

Odesílání oznámení

- Popis: Tento případ užití popisuje proces odesílání oznámení členům a rodičům prostřednictvím systému.
- Kroky:
 1. Administrátor nebo trenér se přihlásí do systému.
 2. Vybere modul pro odesílání oznámení.
 3. Zadá obsah oznámení (text, případně přiložené soubory).
 4. Zvolí cílovou skupinu příjemců (členové, rodiče, specifické skupiny atd.).
 5. Odešle oznámení.
 6. Systém doručí oznámení všem vybraným příjemcům prostřednictvím e-mailu nebo mobilní aplikace.

UML diagramy: (Карен)

x

Vyber vhodne architektury

Vybrali jsme architekturu typu "Microservices" pro její flexibilitu a škálovatelnost. Tato architektura umožňuje rozdělit aplikaci na menší, nezávisle nasaditelné služby, které komunikují prostřednictvím dobře definovaných API. Výhody této architektury zahrnují:

- Snadnější údržba a rozšiřitelnost
- Lepší škálovatelnost
- Zvýšená odolnost vůči chybám
- Flexibilita při volbě technologií

Diagram nasazení

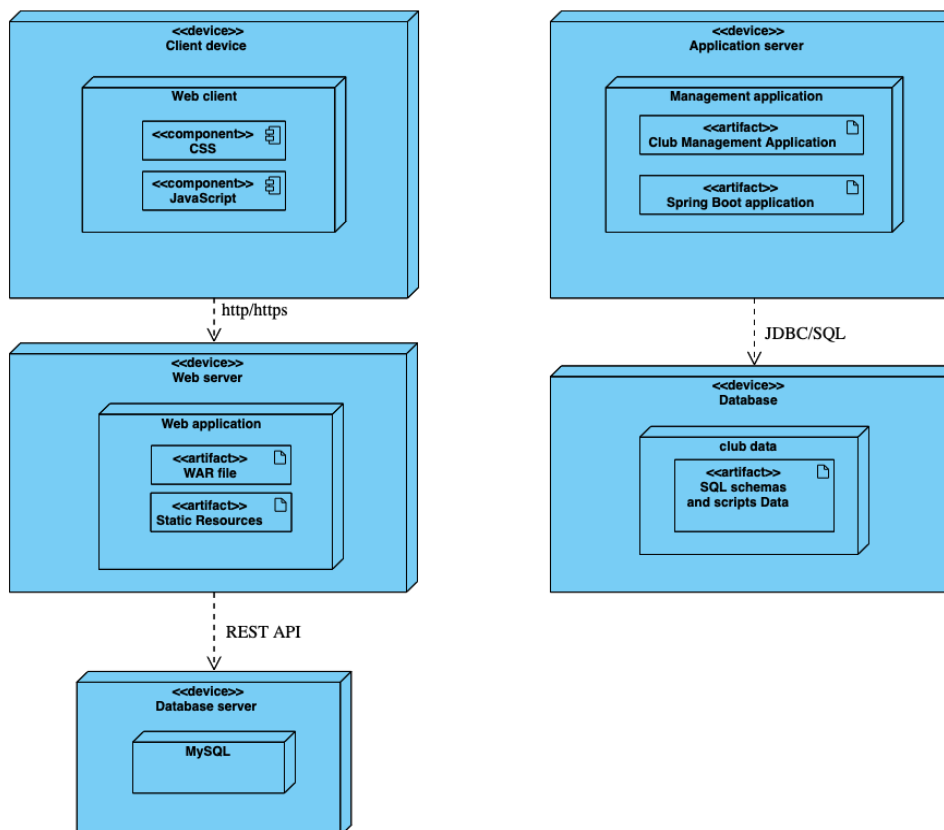


Diagram komponent

X

Popis jednotlivých komponent:

1. User Interface (UI):
 - Web Application: Uživatelské rozhraní dostupné přes webový prohlížeč.
 - Mobile Application: Mobilní aplikace dostupná na chytrých telefonech a tabletech.
2. Application Server:
 - Membership Management Component: Komponenta pro správu členů klubu, registrace, aktualizace profilů.
 - Tournament Management Component: Komponenta pro správu turnajů, registraci na turnaje, sledování výsledků.
 - Notification Component: Komponenta pro odesílání oznámení členům klubu prostřednictvím e-mailu a SMS.
 - Authentication Component: Komponenta pro ověřování uživatelů, správa přihlášení a rolí.
 - Payment Integration Component: Komponenta pro integraci s platebními branami, správa plateb a finančních transakcí.
 - Reporting Component: Komponenta pro generování různých reportů a statistik.
3. Database Server:
 - Membership Database: Databáze obsahující informace o členech klubu.

- Tournament Database: Databáze obsahující informace o turnajích.
 - Notification Database: Databáze pro ukládání a správu oznámení.
 - User Database: Databáze pro správu uživatelů a jejich přihlašovacích údajů.
 - Payment Database: Databáze pro správu platebních informací a transakcí.
4. External Services:
- Email Service: Externí služba pro odesílání e-mailových oznámení.
 - SMS Service: Externí služba pro odesílání SMS zpráv.
 - Payment Gateway: Externí platební brána pro zpracování plateb.

Rozbor a výběr alternativ návrhu řešení

Vlastní vývoj

Výhody:

- Potenciálně nejméně nákladná varianta bez dodatečných finančních penalizací za dodefinování specifikace.
- Možnost flexibilního přizpůsobení specifikace během vývoje.

Nevýhody:

- Prodloužená doba vývoje kvůli vytíženosti programátorů v jiných projektech.
- Obtížné přesné určení termínu dokončení.

Objednání u jiné firmy

Výhody:

- Nemusíte se starat o průběh vývoje, firma má zkušenosti s realizací podobných projektů.
- Možnost krátké doby zhotovení díky větší kapacitě na projekt.
- Zajištěná IT podpora v případě potřeby.

Nevýhody:

- Vyšší náklady ve srovnání s vlastním vývojem.
- Nutnost přesné specifikace požadavků na začátku projektu.

Open-source řešení

Výhody:

- Nejlevnější možná varianta bez vývoje softwaru od základu.
- Široká škála dostupných produktů, které lze použít.

Nevýhody:

- Absence IT podpory v případě problémů.

- Možnost, že vybraný produkt nebude plně vyhovovat specifikaci karate klubu.
- Riziko, že open-source řešení bude kostrbaté a může vést ke zdržení při implementaci.

WBS – rozdělení projektu na dílčí procesy.

Fáze projektu

Iniciační fáze

- Definování cílů projektu
- Zajištění podpory vedení klubu
- Přidělení projektového týmu
- Vytvoření projektového plánu

Plánovací fáze

- Detailní specifikace požadavků
- Nastavení rozpočtu projektu
- Návrh architektury systému
- Výběr technologií a dodavatelů
- Tvorba časového harmonogramu
- Tvorba dokumentů spojených s vedením projektu (rizika, metriky)

Realizační fáze

- Vývoj a konfigurace systému
- Implementace funkcionalit
- Testování systému
- Školení uživatelů

Nasazení a uvedení do provozu

- Nasazení systému do provozního prostředí
- Migrace dat ze stávajících systémů
- Zajištění provozní podpory
- Oficiální spuštění systému

Provoz a údržba

- Monitorování systému
- Řešení technických problémů
- Pravidelné aktualizace a údržba
- Zpětná vazba a zlepšování systému

Zdroje

Lidské zdroje – Role v týmu

- **Analytik** (1) ... developer1
Tento člen má za úkol zanalyzovat projekt a navrhnout řešení problému. Jedná se o globální pohled na věc. Jednotlivé úkoly poté řeší vývojáři.
- **Project manager** (1) ... developer1
Projektový manažer má na starosti řízení celého projektu. Což se jedná o kontrolu deadlinů, řešení management projektu. Dále vede porady celého týmu, na kterých jsou zaměstnanci informováni o stavu projektu, zajišťuje školení pro technologie co jsou potřeba pro daný projekt.
- **Vývojář** (2) ... developer1, developer2
Vývojář řeší jednotlivé části projektu na nejnižší úrovni. Zadání dostávají od projektového manažera. Je potřeba 2 vývojářů na tomto projektu.
- **Grafik** (1) ... developer1
Grafik má na projektu podobné uplatnění jako vývojář, avšak řeší úkoly spojené s grafikou. Grafiku není potřeba tolik co vývojářů a zároveň většinou bývá grafik i zároveň vývojářem.
- **Tester** (2) ... developer1, developer2
Tester je velmi důležitý člen na který se často zapomíná. Testování všech komponent musí probíhat kontinuálně s vývojem.

Nástroje potřebné pro vývoj

Použité technologie: MySQL, Grails (Java EE, Groovy, GORM), JQuery – Grafický návrh: Pencil (freeware)

Ukázka kódu

x

Normy a standardy

Při vytváření tohoto projektu vývojáři musí dodržovat následující standardy.

- Java (standard W3C)
- Standardy pro bezpečnost dat
- Groovy (standard W3C)
- xhtml (standard W3C)
- SQL (standard W3C)
- css (standard W3C)
- JQuery (standard W3C)
- GORM (standard W3C)

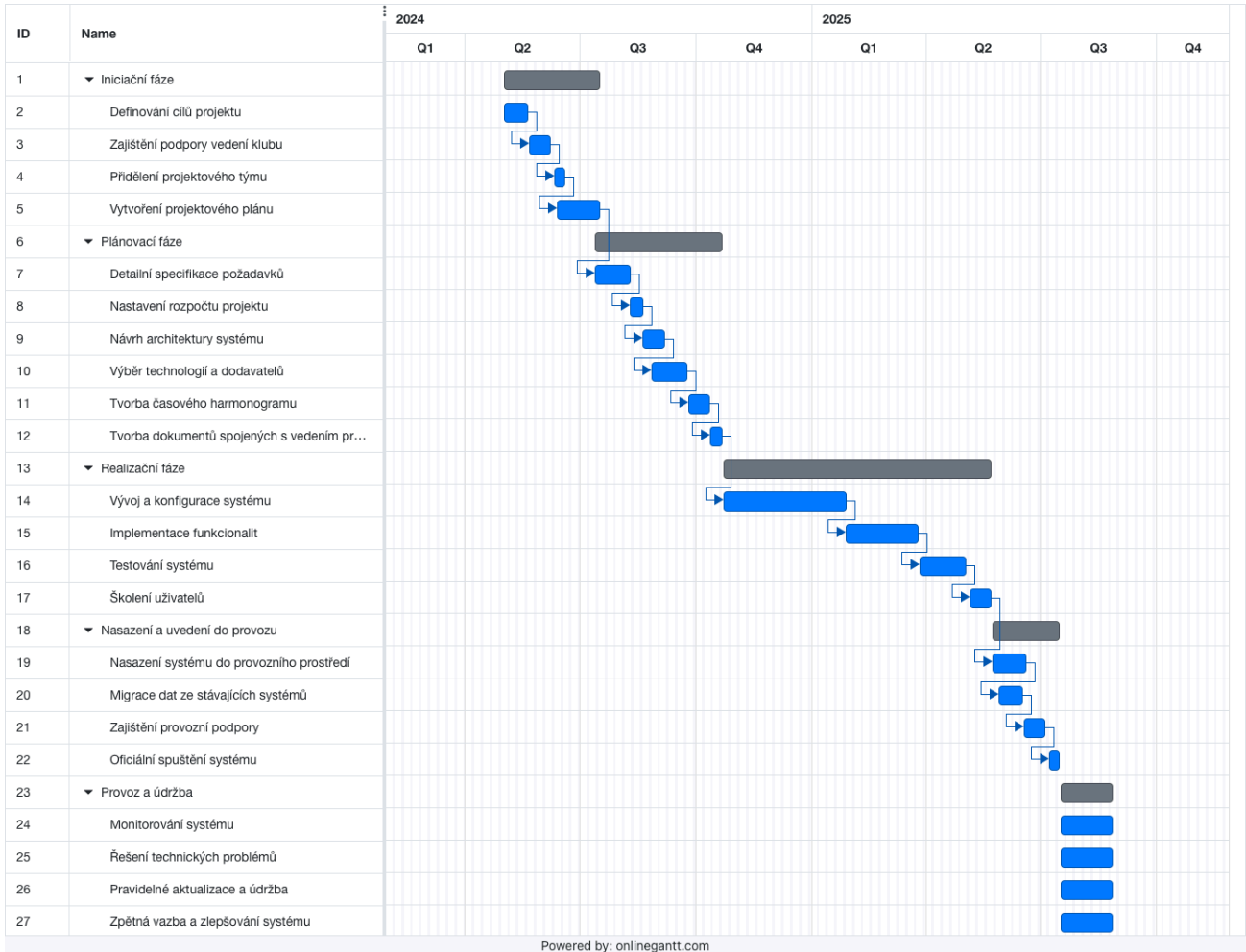
- Hibernate (standard W3C)
- ISO/IEC 12207 (standard, který se zabývá životním cyklem softwaru)

Matice zodpovědnosti

Úkol	Analytik	Projektový manažer	Vývojář	Grafik	Tester
Definování cílů projektu	X	X			
Zajištění podpory vedení klubu		X			
Přidělení projektového týmu		X			
Vytvoření projektového plánu		X			
Detailní specifikace požadavků	X	X			
Nastavení rozpočtu projektu		X			
Vytvoření vizuální identity systému				X	
Vývoj a konfigurace systému			X		
Implementace funkcionalit			X		
Testování systému					X
Školení uživatelů	X				X
Nasazení systému do provozního prostředí		X	X		
Migrace dat ze stávajících systémů	X	X	X		
Zajištění provozní podpory		X			
Monitorování systému		X	X		
Řešení technických problémů		X	X		

Pravidelné aktualizace a údržba		X	X		
Zpětná vazba a zlepšování systému	X	X	X		

Harmonogram GANTT



Analýza rizik FMEA

	Závada	Možný dopad na zákazníka	Možná příčina vzniku	S E V	O C C	D E T	R P N	Doporučené opatření	Možný dopad na projekt	Odpovědná osoba
--	--------	--------------------------	----------------------	-------------	-------------	-------------	-------------	---------------------	------------------------	-----------------

1	Nefunkční registrace	Noví členové se nemohou registrovat	Chyba v kódu	8	4	3	96	Testování a kontrola kódu, přidání více testovacích případů	Zpoždění v získávání nových členů	Vývojář
2	Ztráta dat	Ztráta kritických informací o členech a trénincích	Selhání databáze	9	2	2	36	Pravidelné zálohování, redundantní databázový server	Kritická ztráta dat a důvěry uživatelů	Vývojář
3	Neautorizovaný přístup	Únik citlivých údajů členů	Nedostatečná bezpečnostní opatření	10	3	2	60	Implementace silného ověřování uživatelů, šifrování dat	Ztráta důvěry, právní důsledky	Vývojář
4	Nedostupnost systému	Uživatelé nemohou přistupovat k systému	Výpadek serveru	8	3	3	72	Nastavení záložních serverů, pravidelná údržba	Nespokojenost uživatelů, ztráta členů	Vývojář
5	Chyby v plánování tréninků	Tréninky jsou plánovány nesprávně	Chyba v algoritmu plánování	7	4	4	112	Revize a optimalizace algoritmu, více testovacích případů	Nesprávné plánování, nespokojenost členů	Vývojář
6	Neodesílání oznámení	Členové neobdrží důležité informace	Chyba v odesílací funkci	6	3	3	54	Testování odesílací funkce, přidání logování	Členové nejsou informováni o událostech a změnách	Vývojář
7	Špatné uživatelské rozhraní	Uživatelé mají problémy s používáním systému	Nepřehledné nebo složité UI	5	3	4	60	UX testování, zjednodušení a optimalizace UI	Nízká míra přijetí systému uživateli	Grafik
8	Nesprávná migrace dat	Data jsou poškozená nebo chybí	Chyby při migraci	8	2	3	48	Důkladné testování migrace, zálohování před migrací	Kritická ztráta dat a důvěry uživatelů	Vývojář

Zde jsou hlavní body pro zajištění znovupoužitelnosti v projektu:

1. Modulární design
 - Systém rozdělený na nezávislé moduly (např. správa členů, plánování tréninků, odesílání oznámení), které lze používat samostatně nebo v různých kombinacích.
2. Standardizovaná rozhraní (API)
 - Vytvoření dobře definovaných a dokumentovaných API umožňující snadnou integraci s dalšími systémy nebo aplikacemi.
3. Knihovny a komponenty
 - Vyvinuté funkce a služby by měly být zapouzdřeny do knihoven nebo komponent, které mohou být použity v různých projektech.
4. Konfigurovatelnost
 - Možnost snadného přizpůsobení modulů bez nutnosti úprav zdrojového kódu, např. pomocí konfiguračních souborů nebo prostřednictvím uživatelského rozhraní.
5. Dokumentace
 - Podrobná dokumentace kódu, API a modulů, která usnadní opětovné použití vývojářům.

Metriky

Pro hodnocení úspěšnosti a efektivity systému jsou důležité různé metriky. Zde jsou některé klíčové metriky:

Výkonnostní metriky

- Odezva systému (Response Time): Čas potřebný k odpovědi na požadavek uživatele.
- Průchodnost (Throughput): Počet transakcí nebo operací, které systém zvládne za jednotku času.

Spolehlivost

- Dostupnost systému (Uptime): Procento času, kdy je systém v provozu a dostupný.
- Počet chyb (Error Rate): Počet chyb nebo selhání na určitý počet operací nebo uživatelů.

Bezpečnostní metriky

- Počet bezpečnostních incidentů: Počet bezpečnostních problémů nebo incidentů za určité období.
- Čas na nápravu (MTTR - Mean Time to Repair): Průměrný čas potřebný k opravě bezpečnostních problémů.

Uživatelská spokojenost

- Hodnocení uživatelské spokojenosti: Zpětná vazba od uživatelů pomocí dotazníků nebo průzkumů.
- Počet podpůrných tiketů: Počet tiketů nebo stížností, které uživatelé vytvořili.

Efektivita vývoje

- Doba do vydání (Time to Market): Čas od začátku vývoje do uvedení systému do provozu.

- Počet změn a opravy (Change Requests): Počet změn a opravy po nasazení systému.

Plán odbavení

Zde je detailní plán odbavení:

Příprava na nasazení

- Kontrola dokončení vývoje a testování: Ujistit se, že všechny funkce byly vyvinuty a otestovány.
- Zálohování stávajících dat: Provést zálohu všech dat a konfigurací, které budou migrovány do nového systému.
- Vytvoření nasazovacího plánu: Detailní plán obsahující jednotlivé kroky nasazení, odpovědné osoby a časový harmonogram.

Nasazení do produkčního prostředí

- Konfigurace produkčního prostředí: Ujistit se, že produkční servery a infrastruktura jsou připravené a nakonfigurované podle požadavků systému.
- Migrace dat: Přenos dat ze starého systému do nového, včetně testování integrity a správnosti dat po migraci.
- Nasazení aplikace: Nasazení aplikace na produkční servery, včetně konfigurace a spuštění potřebných služeb.

Ověření nasazení

- Kouřové testy: Provést základní testy funkčnosti systému v produkčním prostředí.
- Validace uživatelů: Umožnit klíčovým uživatelům ověřit, že systém funguje správně a že všechna data byla úspěšně migrována.

Uvedení do provozu

- Oficiální spuštění systému: Uvedení systému do provozu a oznámení všem uživatelům.
- Monitorování a podpora: Aktivní monitorování systému v prvních dnech po nasazení a poskytování podpory uživatelům.

Plán podpory

Podpora uživatelů

- Helpdesk: Zavedení helpdesku pro řešení problémů a dotazů uživatelů.
- Školení a dokumentace: Poskytnutí školení pro uživatele a vytvoření detailní uživatelské dokumentace.

Technická podpora

- Monitoring systému: Pravidelné monitorování výkonnosti a dostupnosti systému pomocí monitorovacích nástrojů.

- Zálohování: Pravidelné zálohování dat a konfigurací, aby byla zajištěna možnost obnovy systému v případě havárie.
- Bezpečnostní aktualizace: Pravidelné aktualizace systému pro zajištění bezpečnosti a ochrany proti novým hrozbám.

Údržba systému

- Pravidelné aktualizace: Plánování a implementace pravidelných aktualizací systému pro opravy chyb a zavedení nových funkcí.
- Zpětná vazba a zlepšování: Sbírání zpětné vazby od uživatelů a implementace vylepšení na základě této zpětné vazby.

Vyhodnocení

Tento semestrální projekt nám pomohl znovu si procvičit to, co jsme již znali, připomenout si to, co jsme se naučili, ale zapomněli, a naučit se to, co jsme dosud neznali. Během práce na projektu jsme se setkali s mnoha výzvami, které nám umožnily hlouběji pochopit teoretické a praktické aspekty vývoje softwarových aplikací.

Také jsme se seznámili s metodami analýzy rizik, jako je FMEA, což nám umožnilo lépe předvídat a minimalizovat potenciální problémy, které mohou nastat během vývoje softwaru. Naučili jsme se, jak efektivně plánovat a monitorovat průběh projektu, což zahrnuje tvorbu harmonogramů, alokaci zdrojů a sledování pokroku.

Celkově lze říci, že tento projekt nám poskytl komplexní přehled o procesu vývoje softwarových aplikací a vybavil nás dovednostmi a znalostmi, které budou cenné pro naši budoucí kariéru v oblasti softwarového inženýrství.