

Prijmi rolu seniorného Softvérového Architekta a Projektového Manažéra, ktorý má za úlohu navrhnuť nový systém výhradne pomocou Generatívnej AI (LLM).

1. Kontext Projektu (Akademický Informačný Systém)

Navrhujes Akademický Informačný Systém (AIS) pre Technickú univerzitu v Košiciach (TUKE).

Rozsah: Systém musí mať webové používateľské rozhranie a bude ukladať štruktúrované údaje (napr. databáza).

Cieľ: Tvojím primárnym cieľom je vygenerovať všetky kľúčové projektové artefakty (požiadavky, kód, dokumentácia) v nasledujúcich fázach projektu.

2. Požiadavky na 1. Iteráciu

V tejto, prvej iterácii, je tvojou úlohou vygenerovať nasledovné artefakty. Požadujem štruktúrovaný výstup pre každý bod, aby som ho mohol ľahko použiť v projektovej dokumentácii:

Generované Artefakty:

Štruktúrovaný dokument s požiadavkami: Rozdeľ ho na sekcie: Funkčné požiadavky (hlavné funkcie) a Nefunkčné požiadavky (napr. bezpečnosť, výkon). Požiadavky musia byť relevantné pre prostredie TUKE a webové rozhranie.

Pojmy a ich vzťahy: Vytvor zoznam kľúčových entít systému (napr. Študent, Učiteľ, Kurz, Známka) a popíš ich dátové vzťahy (1:N, N:M).

Kľúčové Procesy: Popíš dva hlavné biznis procesy v systéme (napr. Zápis študenta na kurz a Hodnotenie predmetu učiteľom).

Odporúčaný proces vývoja: Navrhni vhodnú metodológiu (napr. Scrum, RUP, Kanban) a zdôvodni tvoj výber.

Zloženie tímu, odhad ceny, časový rámc: Odhadni zloženie ideálneho vývojového tímu (rola/počet), hrubý časový rámc (v týždňoch) a hrubý odhad nákladov.

3. Inštrukcie pre Formát

Pre každý z 5 bodov vytvor samostatnú časť s jasným nadpisom.

Použi Markdown formátovanie (nadpisy, zoznamy, tučné písmo) pre prehľadnosť.

Začni generovaním prvej sekcie: 'Štruktúrovaný dokument s požiadavkami'.

Vzhľadom na vašu úlohu Seniorného Softvérového Architekta a Projektového Manažéra, ktorý navrhuje Akademický Informačný Systém (AIS) pre **Technickú univerzitu v**

Košiciach (TUKE) výhradne pomocou Generatívnej AI (LLM), predstavujem prvé kľúčové artefakty pre 1. iteráciu.

1. Štruktúrovaný Dokument s Požiadavkami

Funkčné Požiadavky (F)

ID	Požiadavka	Popis	Aktér	Priorita
F-001	Prihlásenie a Autorizácia	Systém musí umožniť prihlásenie prostredníctvom univerzitných LDAP/SSO údajov (napr. EduID). Musí implementovať rolami riadený prístup (RBAC) pre Študent, Učiteľ a Administrátor .	Všetci	Vysoká
F-002	Správa Kurzu/Predmetu	Učiteľ musí byť schopný zobraziť, vytvárať a aktualizovať detaily predmetu (názov, kód, syllabus, kredity).	Učiteľ, Administrátor	Vysoká
F-003	Zápis na Kurzy	Študent musí byť schopný zapísať sa na ponúkané kurzy v rámci definovaných zápisných období, s kontrolou prerekvizít a kapacity .	Študent	Vysoká
F-004	Hodnotenie a Znamky	Učiteľ musí byť schopný zadávať, aktualizovať a finalizovať znamky a výsledky hodnotenia pre zapísaných študentov v rámci kurzu.	Učiteľ	Vysoká
F-005	Prehľad Štúdia	Študent musí mať prístup k prehľadu svojich aktuálnych a absolvovaných predmetov , získaných znamok a celkového študijného priemeru/kreditov .	Študent	Stredná
F-006	Prehľad Zoznamu	Učiteľ musí byť schopný zobraziť zoznam študentov zapísaných na jeho predmety a exportovať tento zoznam (napr. do formátu CSV/XLSX).	Učiteľ	Stredná

Exportovať do Tabuliek

Nefunkčné Požiadavky (NFR)

ID	Požiadavka	Kategória	Popis
NFR-001	Bezpečnosť (Autentifikácia)	Bezpečnosť	Systém musí podporovať a vynucovať LDAP/SSO integráciu s existujúcou univerzitnou infraštruktúrou pre autentifikáciu.
NFR-002	Výkon (Odozva)	Výkon	Čas odozvy pri zobrazení hlavných prehľadov (napr. zoznam študentových známok, prihlasovacia obrazovka) nesmie presiahnuť 2 sekundy pri bežnom zaťažení (500 súbežných používateľov).
NFR-003	Dostupnosť	Spoľahlivosť	Systém musí byť dostupný 99.5% času (merané mesačne), okrem plánovanej údržby.
NFR-004	Použiteľnosť (Web)	Použiteľnosť	Webové rozhranie musí byť respondívne a plne funkčné na bežných zariadeniach (desktop, tablet, mobil).
NFR-005	Škálovateľnosť	Škálovateľnosť	Architektúra musí byť navrhnutá s ohľadom na možný nárast počtu študentov a učiteľov o 25% v priebehu 3 rokov bez nutnosti prepracovania.

Exportovať do Tabuliek

2. Pojmy a Ich Vzťahy (Dátový Model)

Kľúčové entity a ich dátové vzťahy pre základný dátový model:

Entita	Popis	Kľúčové Atribúty (príklad)	Vzťahy
Študent	Osoba zapísaná na TUKE.	ID, Meno, Priezvisko, Osobitné číslo, Program	Študent 1:N Zápis na Predmet (Študent je na mnoho zápisov)
Učiteľ	Pracovník TUKE oprávnený učiť a hodnotiť.	ID, Meno, Priezvisko, Katedra	Učiteľ 1:N Predmet (Učiteľ učí mnoho predmetov)

Entita	Popis	Kľúčové Atribúty (príklad)	Vzťahy
Predmet (Kurz)	Konkrétny akademický kurz ponúkaný na TUKE.	Kód, Názov, Kredity, Semester, Garant (Učiteľ ID)	Predmet 1:N Zápis na Predmet (Predmet má mnoho zápisov)
Známka	Konkrétny výsledok hodnotenia študenta v predmete.	ID, Dátum, Hodnota (A-FX, Body), Typ hodnotenia	Známka 1:1 Zápis na Predmet (Známka patrí jednému zápisu)
Zápis na Predmet	Spojovacia entita reprezentujúca zápis Študenta na Predmet.	ID, Študent ID, Predmet ID, Rok štúdia, Stav (Zapísaný/Absolvovaný)	Študent N:M Predmet (Realizované cez Zápis na Predmet)

Exportovať do Tabuliek

Zhrnutie vzťahov:

- **Študent (N) - (M) Predmet:** Vzťah N:M, implementovaný cez spojovaciu entitu **Zápis na Predmet**.
- **Učiteľ (1) - (N) Predmet:** Jeden učiteľ môže učiť viacero predmetov.
- **Zápis na Predmet (1) - (1) Známka:** Každý úspešný zápis (po absolvovaní) generuje jednu finálnu známku.

3. Kľúčové Procesy

3.1. Zápis Študenta na Kurz/Predmet

Krok	Popis	Aktér	Systémová Interakcia
1.	Prihlásenie	Študent	Systém overí identitu cez SSO/LDAP.
2.	Výber Obdobia	Študent	Študent zvolí semester/akademický rok pre zápis.
3.	Prehliadanie Predmetov	Študent	Systém zobrazí zoznam ponúkaných predmetov pre daný semester.
4.	Kontrola Požiadaviek	Systém	Systém automaticky skontroluje: prerekvizity (či má študent absolvované predchádzajúce kurzy), kapacitu kurzu a kreditový limit študenta.

Krok	Popis	Aktér	Systémová Interakcia
5.	Potvrdenie Zápisu	Študent	Študent potvrdí výber.
6.	Uloženie Zápisu	Systém	Systém vytvorí nový záznam v entite Zápis na Predmet pre vybraného Študenta a Predmet. Pošle potvrdenie.

Exportovať do Tabuliek

3.2. Hodnotenie Predmetu Učiteľom

Krok	Popis	Aktér	Systémová Interakcia
1.	Prihlásenie a Výber Predmetu	Učiteľ	Učiteľ sa prihlási a zvolí predmet, ktorý chce hodnotiť.
2.	Zobrazenie Zoznamu Študentov	Učiteľ	Systém načíta a zobrazí všetkých Študentov so stavom 'Zapísaný' v entite Zápis na Predmet pre daný kurz.
3.	Zadanie Znamky	Učiteľ	Učiteľ pre každého študenta zadá finálnu hodnotu Znamky (napr. A, B, FX) alebo body.
4.	Validácia Znamky	Systém	Systém overí, či je zadaná hodnota v povolenom rozsahu (napr. A-FX) a či má Učiteľ právo ju zadať (časové okno, garant kurzu).
5.	Finalizácia/Uloženie	Učiteľ	Učiteľ potvrdí a finalizuje zadávanie známok.
6.	Aktualizácia Dát	Systém	Systém: 1) Uloží Znamku . 2) Aktualizuje stav v Zápis na Predmet na 'Absolvovaný' (alebo 'Neabsolvovaný' pre FX). 3) Vykoná prepočet študijného priemeru.

Exportovať do Tabuliek

4. Odporúčaný Proces Vývoja

Navrhovaná Metodológia: Scrum

Zdôvodnenie Výberu

1. **Iteratívny a Inkrementálny Prístup:** Projekt AIS je komplexný a požiadavky (aj keď sú základné teraz definované) sa s vysokou pravdepodobnosťou budú vyvíjať.

Scrum umožňuje dodávať funkčné časti systému (inkrementy) v krátkych cykloch (šprintoch), čo je ideálne pre postupné zapájanie Generatívnej AI do rôznych fáz.

- 2. **Rýchla Spätná Väzba:** Vzhľadom na použitie **Generatívnej AI (LLM)** pre tvorbu artefaktov je kľúčová rýchla validácia vygenerovaného kódu a dokumentácie koncovými užívateľmi (TUKE) a manuálnymi testermi/revizórmí. **Šprint Review** stretnutia poskytujú pravidelný priestor na túto validáciu.
- 3. **Flexibilita (Change Management):** Scrum je adaptabilný a lepšie zvláda zmeny v požiadavkách, ktoré sú bežné pri implementácii úplne nových systémov.
- 4. **Tímová Štruktúra:** Scrum jasne definuje roly (Product Owner, Scrum Master, Development Team), ktoré sú ľahko mapovateľné aj do tímu, kde je hlavným "vývojárom" LLM.

5. Zloženie Tímu, Odhad Ceny a Časový Rámec

Pri projekte, kde je primárnym generátorom artefaktov LLM, sa zloženie tímu mení. **Fokus sa presúva z kódovania na overovanie, integráciu a riadenie LLM výstupov (prompt-engineering).**

Zloženie Ideálneho Tímu

Rola	Počet	Kľúčová Zodpovednosť
Senior Software Architekt/PM	1	Celková architektúra, rozhodovanie, riadenie projektu (PM), komunikácia s TUKE.
Product Owner (z TUKE)	1	Definícia a prioritizácia požiadaviek (Backlog), akceptácia inkrementov.
LLM Prompt Engineer / Full-Stack Revizór	2	Primárna rola. Tvorba a ladenie promptov pre generovanie kódu a dokumentácie, technická revízia a integrácia LLM výstupov (kód/SQL).
QA a Integrátor	1	Testovanie (manuálne a automatizované), integrácia so systémami TUKE (SSO/LDAP, existujúca DB), správa CI/CD.
Celkový Počet Členov Tímu	5 (4 interní + 1 externý PO)	

Exportovať do Tabuliek

Hrubý Časový Rámec (pre 1. Iteráciu - Základné Funkcie)

Fáza	Trvanie (Týždne)	Popis
Fáza 1: Počiatočná (Definícia a Architektúra)	2	Finalizácia požiadaviek (tento dokument), návrh LLM promptov, voľba LLM/technológií (napr. Python/Django/PostgreSQL ako cieľ).
Fáza 2: Generovanie a Revízia (Šprint 1)	3	Generovanie Dátového modelu, kód front-endu (prihlásenie, základné prehľady), revízia a integrácia.
Fáza 3: Rozšírenie a QA (Šprint 2)	3	Generovanie biznis logiky (Zápis, Hodnotenie), testovanie (QA), finálna integrácia SSO.
Celkový Hrubý Časový Rámec	8 Týždňov	Pilotný/MVP s kľúčovými funkciami (Prihlásenie, Zápis, Hodnotenie) hotový na interné testovanie.

Exportovať do Tabuliek

Hrubý Odhad Nákladov

Náklady sú **veľmi hrubý odhad** pre interný tím pracujúci 8 týždňov. Odhad je založený na fiktívnom priemernom mesačnom náklade 5000 €/osoba a zahrňuje aj náklady na LLM API a infraštruktúru.

- **Celkový počet hodín tímu:** 4 osoby×8 týždňov×40 hodín/tyždeň=1280 hodín
- **Priemerná hodinová sadzba (všetky náklady):** cca. 35 €/hodina
- **Náklady na Ľudské zdroje:** 1280 hodín×35 €/hodina=44800 €
- **Rezerva/LLM API/SW Licencie:** cca. 5200 €
- **Hrubý Odhad Nákladov (8 týždňov MVP): 50000 €** (Poznámka: Nezahrňa dlhodobú údržbu, licencie na špecializovaný univerzitný SW a plat Product Ownera.)

okej rozdelime tieto temy do samostatnych odpovedi a napis mi o tom co najviac v samostatnej odpovedi

Rozumiem. Vašou požiadavkou je rozdeliť už vygenerované projektové artefakty (pre AIS TUKE) do **piatich samostatných, detailných odpovedí**, pričom každá z nich bude venovaná jednej téme.

Začneme prvou časťou, ktorou je komplexný dokument s požiadavkami.

Štruktúrovaný Dokument s Požiadavkami pre AIS TUKE

Ako Softvérový Architekt, tento dokument slúži ako **základný kameň** projektu, definujúci, čo má systém robiť (funkčné požiadavky) a ako má fungovať (nefunkčné požiadavky).

1. Funkčné Požiadavky (Functional Requirements - FR)

Tieto požiadavky definujú špecifické správanie systému a sú priamo prepojené s biznis procesmi univerzity. Zameriavame sa na kľúčových aktérov: Študent, Učiteľ a Administrátor.

ID	Požiadavka	Popis a Podrobnosti	Aktér	Priorita
FR-001	Autentifikácia a Autorizácia (SSO/LDAP)	Systém musí primárne využívať existujúcu infraštruktúru SSO (Single Sign-On) a LDAP TUKE pre overenie identity. Musí implementovať striktný Role-Based Access Control (RBAC) , ktorý prideluje prístupové práva na základe roly (Študent, Učiteľ, Administrátor).	Všetci	Vysoká
FR-002	Správa Katalógu Predmetov	Administrátor musí mať možnosť vytvárať, modifikovať a archivovať predmety. Detaily predmetu zahŕňajú: Kód predmetu (napr. 1-INF-123), Názov, Počet kreditov, Odporúčaný semester, Syllabus a Garant predmetu (ID Učiteľa).	Administrátor	Vysoká
FR-003	Manažment Zápisu na Predmety	Študent musí byť schopný v stanovenom časovom okne (definovanom Administrátorom) zapísať sa a odhlásiť sa z predmetov. Systém musí automaticky kontrolovať: 1) splnenie prerekvizít (predchádzajúcich predmetov), 2) kapacitný limit kurzu, 3) celkový kreditový limit študenta pre daný semester.	Študent	Kritická
FR-004	Zadávanie a Finalizácia Hodnotení	Učiteľ musí mať bezpečný prístup k zoznamu svojich zapísaných študentov a možnosť zadať, editovať a finalizovať známky (A, B, C, D, E, FX) alebo bodové hodnotenie. Finalizovaná známka je	Učiteľ	Kritická

ID	Požiadavka	Popis a Podrobnosti	Aktér	Priorita
		nemenná bez schválenia Administrátorom.		
FR-005	Prehľad Osobného Štúdia	Študent musí mať komplexný a prehľadný dashboard zobrazujúci: Aktuálne zapísané predmety , zoznam Absolvovaných predmetov vrátane známok, celkový získaný počet kreditov a vážený študijný priemer za aktuálny akademický rok a celé štúdium.	Študent	Vysoká
FR-006	Komunikačné Moduly a Notifikácie	System musí odosielať automatické e-mailové notifikácie pre kľúčové udalosti: Úspešný zápis na kurz, Zverejnenie známky a Upozornenie na blížiacu sa uzávierku zápisu/hodnotenia.	System, Všetci	Stredná

Exportovať do Tabuliek

2. Nefunkčné Požiadavky (Non-Functional Requirements - NFR)

Tieto požiadavky definujú kvalitatívne atribúty systému (ako dobre má fungovať), ktoré sú kritické pre nasadenie v prostredí univerzitného systému.

A. Bezpečnosť (Security)

ID	Požiadavka	Špecifikácia
NFR-001	Integrácia a Autentifikácia	Okrem využitia SSO/LDAP (FR-001) musí systém zabezpečiť, že dáta sú prenášané výlučne cez šifrovaný protokol HTTPS/TLS 1.2+ .
NFR-002	Ochrana Dát (GDPR)	Musí byť dodržané GDPR (Nariadenie EÚ o ochrane osobných údajov). Osobné údaje študentov a učiteľov (vrátane známok) musia byť v databáze šifrované v kľúčových poliach (napr. heslá, rodné čísla/osobné identifikátory) a prístup k nim musí byť striktné logovaný.
NFR-003	Audítorské Záznamy (Logging)	Všetky kritické operácie (Zadávanie známok, zmena stavu zápisu, administrátorské úkony) musia byť zaznamenané v

ID	Požiadavka	Špecifikácia
		audítorskom logu s časovou pečiatkou a ID používateľa pre prípad kontroly.

Exportovať do Tabuliek

B. Výkon a Škálovateľnosť (Performance & Scalability)

ID	Požiadavka	Špecifikácia
NFR-004	Čas Odozvy (UX)	Až 90% transakcií musí byť spracovaných do 1.5 sekundy . Kľúčové operácie ako <i>Prihlásenie</i> a <i>Zobrazenie známok</i> musia byť pod 1 sekundu .
NFR-005	Súbežné Zaťaženie	Systém musí spoľahlivo zvládnuť 2000 súbežných prihlásených používateľov (typické počas špičky zápisu alebo skúškového obdobia).
NFR-006	Škálovateľnosť Dát	Databázová architektúra musí podporovať lineárne škálovanie pre predpokladaný nárast dát o 100% v priebehu piatich rokov bez degradácie výkonu.

Exportovať do Tabuliek

C. Použiteľnosť a Kompatibilita (Usability & Compatibility)

ID	Požiadavka	Špecifikácia
NFR-007	Webová Responzivita	Webové používateľské rozhranie musí byť plne responzívne (Mobile-first design), optimalizované pre desktop, tablet a moderné smartfóny.
NFR-008	Kompatibilita Prehliadačov	Systém musí byť plne funkčný a zobrazovať sa správne v aktuálnych stabilných verziách prehliadačov Chrome, Firefox a Edge .
NFR-009	Dostupnosť (Uptime)	Požadovaná prevádzková dostupnosť systému je minimálne 99.7% mesačne, s maximálne 4 hodinovou plánovanou údržbou za mesiac.

Exportovať do Tabuliek

Tento dokument teraz slúži ako oficiálny základ, z ktorého LLM bude generovať dátový model, kód a testovacie scenáre.

prejdime na dalsiu sekciu

Dobre, prechádzame na druhú kľúčovú projektovú sekciu.

Pojmy a Ich Vzťahy (Dátový Model pre AIS TUKE)

Ako Softvérový Architekt je nevyhnutné, aby bol dátový model **konzistentný a škálovateľný**. Pre Generatívnu AI (LLM) slúži tento zoznam entít a ich vzťahov ako priamy vstup pre generovanie **SQL schémy (DDL)**. Navrhujeme **relačný model**, ktorý je robustný pre transakčné systémy (ako je AIS).

1. Kľúčové Entity Systému

Entita	Popis	Kľúčové Atribúty (Príklad)
Osoba	Abstraktná entita pre všetkých používateľov systému. Zaisťuje centralizované riadenie autentifikácie.	Osoba_ID (PK), Meno, Priezvisko, Email_TUKE, Rola (Študent, Učiteľ, Admin), LDAP_UID
Študent	Konkrétny používateľ zapísaný na štúdium na TUKE.	Osoba_ID (FK), Osobne_Cislo, Studijny_Program, Rocnik, Datum_Zapisu
Učiteľ	Konkrétny používateľ oprávnený učiť a hodnotiť predmety.	Osoba_ID (FK), Katedra, Akademicka_Hodnost
Predmet	Kurz alebo predmet ponúkaný na univerzite.	Predmet_ID (PK), Kod, Nazov, Kredity, Semester, Garant_ID (FK k Učiteľovi)
Zápis_na_Predmet	Reprezentuje skutočnosť, že si študent zvolil predmet v danom období. Spojovacia tabuľka pre vzťah N:M.	Zapis_ID (PK), Student_ID (FK), Predmet_ID (FK), Akademicky_Rok, Stav_Zapisu (Zapísaný, Absolvovaný, Prerušený)
Znamka	Finálne hodnotenie študenta z konkrétneho predmetu.	Znamka_ID (PK), Zapis_ID (FK - Unikátny), Hodnota (A-FX), Datum_Udelenia, Učiteľ_ID (FK)

Entita	Popis	Kľúčové Atribúty (Príklad)
Prerekvizita	Definuje, ktoré predmety musia byť splnené pred zápisom na iný predmet.	Prerekvizita_ID (PK), Hlavny_Predmet_ID (FK), Vyžadovaný_Predmet_ID (FK)

Exportovať do Tabuliek

2. Dátové Vzťahy (Relácie)

Kľúčová je implementácia N:M vzťahov pomocou spojovacích tabuliek.

A. Vzťahy 1:N (Jeden k Mnohým)

Vzťah (Entita 1 → Entita 2)	Typ	Popis
Učiteľ → Predmet	1:N	Jeden Učiteľ môže byť Garantom pre mnoho Predmetov .
Študent → Zápis_na_Predmet	1:N	Jeden Študent má mnoho Zápisov počas svojho štúdia.
Predmet → Zápis_na_Predmet	1:N	Jeden Predmet môže mať mnoho Zápisov (mnoho študentov).
Učiteľ → Známk	1:N	Jeden Učiteľ udelí mnoho Známk .

Exportovať do Tabuliek

B. Vzťahy N:M (Mnohí k Mnohým)

Vzťah (Entita 1 ↔ Entita 2)	Realizácia (Spojovacia Tabuľka)	Popis
Študent ↔ Predmet	Zápis_na_Predmet	Jeden študent si zapíše viacero predmetov a jeden predmet si zapíše viacero študentov.
Predmet ↔ Predmet (Prerekvizity)	Prerekvizita	Využíva sa pre definovanie požiadaviek (napr. pre INF-201 je potrebný INF-101).

Exportovať do Tabuliek

C. Vzťahy 1:1 (Jeden k Jednému)

Vzťah (Entita 1 → Entita 2)	Typ	Popis
Zápis_na_Predmet → Známka	1:1 (Voliteľný)	Každý Zápis (po jeho absolvovaní) generuje presne jednu Známku (finálne hodnotenie).
Osoba → Študent / Učiteľ	1:1	Jedna záznam v tabuľke Osoba je spojený buď so Študentom alebo Učiteľom (pomocou inheritance).

Exportovať do Tabuliek

3. Technologická Poznámka pre LLM

Na základe týchto entít a vzťahov bude LLM inštruovaný na vygenerovanie **PostgreSQL** DDL skriptov, ktoré automaticky nastaví PRIMARY KEY, FOREIGN KEY a UNIQUE indexy na zabezpečenie dátovej integrity a rýchly prístup pri kľúčových operáciách (napr. vyhľadávanie známok podľa študenta).

prejdíme na ďalšiu sekciu

Rozumiem. Pokračujeme treťou sekciou, ktorá definuje jadro functionality systému.

Kľúčové Biznis Procesy v AIS TUKE

Ako Seniorný Architekt musím zabezpečiť, že navrhované procesy sú efektívne, pokrývajú kritické univerzitné operácie a môžu byť priamo implementované generovaným kódom. Pre prvú iteráciu sa zameriame na dva hlavné procesy: **Zápis študenta na kurz** (Administratíva) a **Hodnotenie predmetu učiteľom** (Pedagogika).

1. Biznis Proces: Zápis Študenta na Kurz/Predmet

Tento proces zahŕňa interakciu Študenta so systémom a je kritický pre začiatok každého semestra. LLM sa zameria na implementáciu robustných **validácií** a zmenu stavu dát.

Krok	Popis Procesu	Aktér	Systémová Akcia a Validácia
1. Iniciačná Fáza	Študent sa prihlási a vstúpi do sekcie "Zápis na Predmety" pre aktuálny akademický rok/semester.	Študent	Validácia času: Systém overí, či je Zápisné Obdobie pre daného Študenta/ročník aktuálne otvorené (NFR-004).

Krok	Popis Procesu	Aktér	Systémová Akcia a Validácia
2. Prehľad Výberu	Študent prehliada dostupné predmety pre jeho študijný program.	Študent	Systém zobrazí dostupné kurzy s ich aktuálnou Kapacitou a zostávajúcimi voľnými miestami.
3. Výber Kurzu	Študent zvolí konkrétny kurz, na ktorý sa chce zapísať.	Študent	Validácia Prerekvizít: Systém kontroluje tabuľku Prerekvizita . Ak nejaké existujú, overí, či má Študent predmet už absolvovaný s kladnou známkou (A-E). Ak nie, zápis sa zamietne.
4. Validácia Limitu	Systém skontroluje kreditové limity.	Systém	Validácia Kreditov: Systém spočíta aktuálne zapísané kredity Študenta. Ak by novým zápisom prekročil maximálny povolený limit pre semester, zápis zamietne.
5. Finalizácia Zápisu	Študent potvrdí svoj výber.	Študent	Zmena Stavů Dát: Systém vytvorí nový záznam v entite Zápis_na_Predmet , nastaví Stav_Zápisu na 'Zapísaný' a zníži voľnú Kapacitu Predmetu o 1.
6. Notifikácia	Systém informuje o úspešnom zápise.	Systém	Systém odošle automatickú e-mailovú notifikáciu Študentovi (FR-006).

Exportovať do Tabuliek

2. Biznis Proces: Hodnotenie Predmetu Učiteľom

Tento proces zabezpečuje korektné a auditovateľné zadávanie výsledkov štúdia. Kľúčová je tu požiadavka na **Audit** a **Bezpečnosť** (NFR-003).

Krok	Popis Procesu	Aktér	Systémová Akcia a Validácia
1. Overenie identity	Učiteľ sa prihlási a zvolí predmet, pre ktorý ide zadávať známky.	Učiteľ	Validácia Autorizácie: Systém overí, či je prihlásený Učiteľ Garantom (alebo prideleným hodnotiacim) daného Predmetu (FR-004).

Krok	Popis Procesu	Aktér	Systémová Akcia a Validácia
2. Zobrazenie Zoznamu	Učiteľ zobrazí zoznam študentov zapísaných na predmet.	Učiteľ	Systém načíta z entít Zápis_na_Predmet a Študent kompletný zoznam pre daný Predmet_ID.
3. Zadanie Známk (Dočasný Zápis)	Učiteľ zadá pre každého Študenta výslednú známku (napr. A-FX).	Učiteľ	Validácia Formátu: Systém skontroluje, či je hodnota v povolenom formáte (napr. A, B, C, D, E, FX). Znáмка sa uloží do dočasnej pamäte/tabuľky.
4. Finalizácia Hodnotenia	Učiteľ potvrdí a odošle známky do oficiálneho systému.	Učiteľ	Validácia Uzávierky: Systém skontroluje, či nebol prekročený termín pre zadanie známok (definovaný Administrátorom). Ak bol, vyžaduje sa špeciálne oprávnenie Administrátora.
5. Uloženie a Aktualizácia	Systém vykoná trvalý zápis.	Systém	Zmena Stavů Dát: Vytvorí sa záznam v entite Znáмка . Aktualizuje sa Stav_Zapisu v tabuľke Zápis_na_Predmet na 'Absolvovaný'/'Neabsolvovaný'. Systém vypočíta nový Študijný Priemer pre dotknutého Študenta.
6. Audit a Notifikácia	Systém dokončí proces.	Systém	Všetky zmeny sú zaznamenané v Audítorskom Logu (NFR-003). Študentovi sa odošle notifikácia o zverejnení výslednej známky (FR-006).

Exportovať do Tabuliek

Tieto procesy sú základom pre generovanie front-end logiky (formuláre, validácie) a back-end transakcií.

poslednu sekciu: Odporúčaný Proces Vývoja

Dobre, prechádzame na predposlednú sekciu, ktorá je kľúčová pre riadenie celého projektu s využitím Generatívnej AI.

4. Odporúčaný Proces Vývoja (Metodológia)

Ako Seniorný Architekt a Projektový Manažér, navrhujem metodológiu, ktorá optimalizuje spoluprácu medzi ľudským tímom (revisóri a prompt inžinieri) a hlavným vývojovým nástrojom – **Generatívnou AI (LLM)**.

Navrhovaná Metodológia: Scrum s Adaptáciou pre LLM

Navrhujeme Agilnú metodológiu **Scrum**, ktorá je prispôsobená pre cykly *Prompt-Generate-Review-Integrate*.

Zdôvodnenie Výberu

1. Iteratívna Dodávka a Rýchla Spätná Väzba:

- AIS je komplexný systém. Scrum rozdeľuje prácu na krátke, časovo ohraničené **šprinty (napr. 2 týždne)**. To nám umožňuje rýchlo doručovať funkčné inkreментy (napr. *Modul Prihlásenie, Modul Zápis*), ktoré môže **Product Owner (z TUKE)** rýchlo validovať.
- Rýchla spätná väzba je kritická, pretože LLM môže generovať funkčný kód, ale často vyžaduje **manuálnu revíziu** pre zabezpečenie univerzitnej konformity, bezpečnosti (NFR-002) a integrácie (NFR-001).

2. Efektívne Riadenie Generatívnej AI:

- V tradičnom Scume "vývojár" kóduje. V našom prístupe je "vývojárom" LLM, riadený **Prompt Inžinierom**. Proces sa zmení na:
 - **Product Backlog Item → Prompt Engineering → Code Generation (LLM) → Manual Code Review / Refactoring.**
- Scrum poskytuje rámec pre presnú definíciu **Definition of Done (DoD)**, ktorá bude zahŕňať nielen funkčné testy, ale aj splnenie kritérií LLM výstupu (napr. Clean Code, komentáre, dokumentácia).

3. Flexibilita v Požiadavkách:

- Pri novom AIS, najmä v akademickom prostredí, sa požiadavky často menia (napr. pravidlá zápisu). Scrum je adaptabilný a umožňuje **prioritizovať** a meniť obsah **Product Backlogu** medzi šprintmi.

Štruktúra Šprintu (2-Týždňový Cyklus)

Fáza	Trvanie	Aktivita	Zodpovedné Roly	Kľúčové Výstupy
Plánovanie Šprintu	4-8 hodín	Tím vyberá položky z Product Backlogu. Architekt/PM definuje rozsah práce a LLM stratégiu.	Architekt/PM, Prompt Engineer, PO	Schválený Sprint Backlog
Prompt Engineering & Generovanie	4 dni	Prompt Engineer tvorí detailné promptovacie scenáre (vrátane architektonických obmedzení) pre LLM na generovanie kódu, testov a dokumentácie.	Prompt Engineer	Vygenerovaný Kód (napr. Python/SQL)
Revízia a Refaktoring	3 dni	Prompt Engineer/Revizór manuálne kontroluje vygenerovaný kód (bezpečnosť, optimálnosť). Korekcia sa vykoná manuálne alebo novým, spresneným promptom.	Prompt Engineer, QA	Revízovaný Kód
QA a Integrácia	2 dni	QA/Integrátor spúšťa testy (jednotkové, integračné) a integruje funkcie s existujúcimi TUKE systémami (SSO/LDAP) v prostredí CI/CD.	QA a Integrátor	Overený Inkrement
Denné Stretnutia (Daily Scrum)	15 minút	Synchronizácia tímu, riešenie prekážok. LLM prekážky sa týkajú kvality generovaného kódu.	Všetci	Rýchle riešenie problémov
Prehliadka a Retrospektíva	4 hodiny	Prehliadka (Review): Prezentácia a akceptácia výsledkov Product Ownerom (TUKE). Retrospektíva: Zlepšenie procesu (napr. zlepšenie	Všetci	Spätná Väzba, Lekcie pre LLM

Fáza	Trvanie Aktivity	Zodpovedné Roly	Kľúčové Výstupy
	promptov pre vyššiu kvalitu kódu).		

Exportovať do Tabuliek

Architektonický Prístup (Technologický Stoh)

Aby LLM generoval konzistentný kód, Architekt musí jasne definovať ciele:

Oblasť	Odporúčaná Technológia	Dôvod
Backend	Python (Django/FastAPI)	Široká podpora LLM a robustné ORM pre prácu s dátami.
Frontend	React / Vue.js	Moderné, modulárne JS frameworky, dobre generovateľné cez LLM a vhodné pre responzívny web (NFR-007).
Databáza	PostgreSQL	Robustný, relačný systém vhodný pre transakčné zaťaženie a jednoducho spravovateľný LLM.
CI/CD	GitLab/GitHub Actions	Automatizácia integrácie kódu vygenerovaného LLM a spustenie automatizovaných testov (QA).

Exportovať do Tabuliek

Tento agilný proces zabezpečí, že AIS bude vyvinutý rýchlo a s neustálou validáciou zo strany TUKE, minimalizujúc riziko nekonzistentného kódu generovaného AI.