

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 rámec: Odhadni zloženie ideálneho vývojového tímu (rola/počet), hrubý časový rámec (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'.

Áno, prijímam túto výzvu. Ako seniorný Softvérový Architekt a Projektový Manažér navrhmem Akademický Informačný Systém (AIS) pre TUKE s využitím Generatívnej AI. Toto je výstup prvej iterácie.

1. Štruktúrovaný dokument s požiadavkami

Tento dokument definuje požiadavky na prvú verziu Akademického informačného systému (AIS) pre Technickú univerzitu v Košiciach.

1.1 Funkčné požiadavky (Hlavné funkcie)

- **FR1: Správa používateľských účtov a rolová autentizácia**
 - **FR1.1:** Systém musí podporovať rôzne typy používateľov: Študent, Vyučujúci, Administrátor.
 - **FR1.2:** Používatelia sa musia prihlásiť pomocou univerzitných prihlasovacích údajov (napr. Single Sign-On - SSO).
 - **FR1.3:** Každá rola má špecifické oprávnenia a prístup k rôznym častiam systému.

- **FR2: Správa študijných programov a kurzov**
 - **FR2.1:** Administrátori musia vedieť vytvárať, upravovať a archivovať študijné programy a kurzy.
 - **FR2.2:** Každý kurz musí mať priradené atribúty: kód, názov, kredity, vyučujúceho, rozvrh hodín, kapacitu.
 - **FR2.3:** Systém musí poskytovať verejný katalóg kurzov s možnosťou filtrovania.
- **FR3: Proces zápisu na kurzy**
 - **FR3.1:** Študenti musia vidieť dostupné kurzy pre nadchádzajúci semester.
 - **FR3.2:** Študenti sa môžu zapísať na kurzy, pokiaľ kapacita a predpoklady (napr. absolvované predmety) sú splnené.
 - **FR3.3:** Systém musí poskytovať potvrdenie o zápise a aktuálny prehľad zapísaných kurzov.
- **FR4: Správa hodnotenia a klasifikácie**
 - **FR4.1:** Vyučujúci musia vedieť zadávať a upravovať známky pre študentov zapísaných v ich kurzoch.
 - **FR4.2:** Študenti musia mať prehľad o svojich známkach z jednotlivých kurzov.
 - **FR4.3:** Systém musí automaticky počítať vážený priemer na základe kreditov a znáмок.
- **FR5: Personalizovaný rozvrh hodín**
 - **FR5.1:** Systém musí automaticky generovať osobný rozvrh hodín pre študentov a vyučujúcich na základe zapísaných/vyučovaných kurzov.
 - **FR5.2:** Rozvrh musí byť zobraziteľný v týždennom pohľade.
- **FR6: Vyhľadávanie a reporty**
 - **FR6.1:** Všetci používatelia musia mať možnosť vyhľadávať v katalógu kurzov.
 - **FR6.2:** Administrátori musia mať možnosť generovať reporty (napr. zoznam študentov v kurze, prehľad znáмок).

1.2 Nefunkčné požiadavky

- **NR1: Bezpečnosť**

- **NR1.1:** Všetky citlivé údaje (osobné údaje, známky) musia byť šifrované počas prenosu (HTTPS) a v pokoji.
 - **NR1.2:** Systém musí zabrániť neoprávnenému prístupu (napr. študent nemôže meniť známky).
 - **NR1.3:** Musí byť implementované pravidelné zálohovanie dát.
 - **NR2: Výkon a škálovateľnosť**
 - **NR2.1:** Čas odozvy systému pre väčšinu operácií (načítanie stránky, zápis) by nemal presiahnuť 2 sekundy pri štandardnom zaťažení (súčasne prihlásených 20% používateľov TUKE).
 - **NR2.2:** Architektúra systému musí umožňovať horizontálne škálovanie pre budúce rozšírenia.
 - **NR3: Použiteľnosť (User Experience - UX)**
 - **NR3.1:** Webové rozhranie musí byť responzívne a použiteľné na zariadeniach s rôznou veľkosťou obrazovky (PC, tablet, mobil).
 - **NR3.2:** Rozhranie musí byť lokalizované do slovenského jazyka.
 - **NR4: Kompatibilita**
 - **NR4.1:** Systém musí byť prístupný prostredníctvom moderných webových prehliadačov (Chrome, Firefox, Edge - posledné 2 verzie).
 - **NR4.2:** Musí byť poskytnuté REST API pre potenciálnu integráciu s existujúcimi systémami TUKE (napr. knižnica, ekonomický systém).
-

2. Pojmy a ich vzťahy

Kľúčové entity systému AIS a ich dátové vzťahy:

- **Používateľ (User)**
 - *Atribúty:* ID, Meno, Priezvisko, Email, Rola (Študent/Vyučujúci/Admin)
 - *Vzťahy:*
 - Je generalizáciou entít Študent a Vyučujúci (1:1 s každou).
 - Môže mať 1 alebo viac **Rol** (N:M, cez priradenú tabuľku User_Roles).
- **Študent (Student)**
 - *Atribúty:* Študijné číslo, Rok nástupu, Aktuálny semester

- *Vzťahy:*
 - Je zapísaný do **Kurzov** (N:M, cez entitu **Zápis/Enrollment**).
 - Má priradený 1 **Študijný program** (N:1).
- **Vyučujúci (Teacher)**
 - *Atribúty:* Katedra, Vedecká hodnosť
 - *Vzťahy:*
 - Vyučuje 1 alebo viac **Kurzov** (1:N).
- **Kurz (Course)**
 - *Atribúty:* Kód, Názov, Počet kreditov, Typ (povinný/voliteľný), Kapacita
 - *Vzťahy:*
 - Je vyučovaný 1 **Vyučujúcim** (N:1).
 - Patrí do 1 **Študijného programu** (N:1, ale jeden kurz môže byť v viacerých programoch -> v skutočnosti N:M).
 - Má viacero **Zápisov** (1:N).
- **Študijný program (Study Program)**
 - *Atribúty:* Kód, Názov, Stupeň (Bc./Ing./PhD.)
 - *Vzťahy:*
 - Má priradených viacero **Študentov** (1:N).
 - Obsahuje viacero **Kurzov** (N:M, cez priradenú tabuľku Program_Courses).
- **Zápis (Enrollment)**
 - *Atribúty:* ID zápisu, Dátum zápisu, Finálna známka
 - *Vzťahy:*
 - Spája entitu **Študent** a **Kurz** (je tzv. "prepojovacia entita" pre vzťah N:M).
 - Môže mať viacero **Hodnotení** (1:N).
- **Hodnotenie (Grade)**
 - *Atribúty:* Typ hodnotenia (skúška, test), Body, Dátum
 - *Vzťahy:*

- Patrí k 1 **Zápisu** (N:1).

3. Klúčové Procesy

3.1 Proces zápisu študenta na kurz

1. **Spustenie:** Študent sa prihlási do systému a vyberie možnosť "Zápis na semester".
2. **Overenie:** Systém overí, či študent splnil predpoklady pre zápis (napr. má zaplatené školené, nemá blokáciu).
3. **Zobrazenie ponuky:** Systém zobrazí študentovi katalóg kurzov dostupných pre jeho študijný program a semester, s informáciami o voľných miestach.
4. **Výber a potvrdenie:** Študent vyberie kurzy a pošle žiadosť o zápis.
5. **Validácia:** Systém validuje žiadosť (kapacita kurzu, časové kolízie v rozvrhu, predpoklady).
6. **Uloženie:** Ak validácia prebehne úspešne, systém vytvorí záznamy **Zápis** a aktualizuje rozvrh študenta.
7. **Výstup:** Študent obdrží potvrdenie o úspešnom zápise a aktualizovaný rozvrh.

3.2 Proces hodnotenia predmetu učiteľom

1. **Spustenie:** Vyučujúci sa prihlási do systému a vo svojom rozhraní vyberie konkrétny kurz.
2. **Výber študenta:** Vyučujúci vyberie zo zoznamu študentov zapísaných do kurzu toho, ktorého chce ohodnotiť.
3. **Zadanie hodnotenia:** Vyučujúci vyplní formulár so známkou, typom hodnotenia (napr. "zápočet") a voliteľným komentárom.
4. **Validácia a uloženie:** Systém overí formát známky a uloží ju ako záznam **Hodnotenie** priradený k príslušnému **Zápisu** študenta.
5. **Prepočítanie finálnej známky:** Ak bolo zadané rozhodujúce hodnotenie (napr. skúška), systém automaticky prepočíta a uloží finálnu známku do entity **Zápis**.
6. **Notifikácia:** Študent je automaticky notifikovaný o novej známke (napr. cez email alebo prihlásení do systému).

4. Odporúčaný proces vývoja

Odporúčaná metodológia: Adaptívny hybrid metód Agile (Scrum) a V-model pre regulované prostredie.

Zdôvodnenie výberu:

- **Agile/Scrum:** Táto časť sa použije pre vývoj webového rozhrania a jadra aplikácie. Umožňuje rýchlu feedback loop s budúcimi používateľmi (študentmi, učiteľmi) prostredníctvom krátkych šprintov (2-3 týždne). Flexibilita Agile je kľúčová pre priebežné doladzovanie požiadaviek, ktoré sú v akademickom prostredí často nejednoznačné.
- **V-model (pre kritické časti):** Pre integráciu s existujúcimi systémami TUKE (napr. SSO, databáza študentov) a pre implementáciu bezpečnostných a dátových požiadaviek sa odporúča prísnejší prístup typický pre V-model. Toto zabezpečí, že pre tieto kritické komponenty bude dôkladne otestovaná zhoda špecifikácií s implementáciou, čo je dôležité pre stabilitu a bezpečnosť.
- **Dôvody:**
 1. **Rovnováha medzi flexibilitou a robustnosťou:** Scrum rieši potrebu adaptácie, kým V-model zabezpečí kvalitu kľúčových, menej meniacich sa častí.
 2. **Vhodnosť pre vysoké školy:** Univerzitné prostredie vyžaduje istú mieru formality (ako V-model), ale zároveň potrebuje systém, ktorý reálne vyhovuje používateľom (ako Agile).
 3. **Riziká:** Tento hybrid minimalizuje riziko vzniku systému, ktorý je technicky správny, ale nepoužiteľný (čistý V-model), alebo systému, ktorý je chaotický a nebezpečný (čistý Agile bez disciplíny).

5. Zloženie tímu, odhad ceny, časový rámec

Odporúčané zloženie tímu:

- **Projektový Manažér / Product Owner:** 1 (definuje priority, komunikácia s klientom - TUKE)
- **Business Analytik:** 1 (spolupracuje na špecifikácii detailných požiadaviek)
- **Softvérový Architekt:** 1 (návrh celkovej architektúry, technických špecifikácií)
- **Senior Backend vývojár:** 2 (vývoj serverovej logiky, API, databázy)
- **Senior Frontend vývecklper:** 2 (vývoj responzívneho webového UI)
- **QA Inžinier / Tester:** 1 (tvorba a vykonávanie testov)

- **DevOps Inžinier:** 1 (nastavenie CI/CD, deployment, monitorovanie)

Celkom: 9 členov tímu.

Hrubý časový rámec (pre 1. iteráciu / MVP - Minimálne uskutočniteľný produkt):

- **Fáza špecifikácie a plánovania:** 2 týždne
- **Šprint 0 (setup prostredia, architektúra):** 2 týždne
- **Vývojové šprinty (cca 6 šprintov po 2 týždne):** 12 týždňov
- **Fáza testovania a nasadenia (UAT, deployment):** 3 týždne
- **Rezerva:** 2 týždne

Celkový odhadovaný čas: 21 týždňov (približne 5 kalendárnych mesiacov).

Hrubý odhad nákladov:

- *Predpoklad:* Priemerná denná sadzba na osobu: 400 € (brutto, vrátane réžií).
- *Počet osobodní:* 9 osôb * 21 týždňov * 5 dní = 945 osobodní.
- **Odhadovaná celková cena za vývoj MVP: 945 os.dní * 400 €/deň = 378 000 €.**
- *Poznámka:* Táto suma nezahŕňa licenčné poplatky za softvér/hardvér, náklady na prevádzku a budúcu údržbu systému. Ide o hrubý odhad vývojových nákladov na základe predpokladaného tímu a času.

This response is AI-generated, for reference only.