

Software Requirements Specification (SRS)

Pre projekt: PayMySportClub – Systém na správu členských poplatkov

Tím: UCM FPV – Tím 8

Produktový vlastník: Zuzana Butašová – zároveň tajomníčka klubu

Technológie: Laravel, HTML/CSS/JS, MySQL/PostgreSQL, Fio API, Jotform API, GitHub

Úvod

Cieľom tohto dokumentu je špecifikovať požiadavky na vývoj informačného systému PayMySportClub. Systém má uľahčiť tajomníkovi klubu správu členských príspevkov, evidenciu platieb a notifikáciu členov. Členom má sprístupniť prehľad o predpísaných platbách a ich úhradách.

Účel dokumentu

Tento dokument popisuje všetky funkčné a nefunkčné požiadavky systému, jeho kontext, používateľov a obmedzenia. Slúži ako základ pre návrh, implementáciu a testovanie systému.

Rozsah systému

Systém bude slúžiť na evidenciu členov športového klubu, správu predpísaných a zaplatených platieb, automatizované párovanie platieb podľa variabilného symbolu a zasielanie emailových notifikácií.

Celkový opis systému

PayMySportClub je webová aplikácia určená pre športový klub, ktorý chce zautomatizovať proces správy členských poplatkov.

Používatelia systému

- **Tajomník klubu** – administrátor systému, ktorý spravuje členov, vytvára predpisy a kontroluje úhrady.
- **Manažér klubu** – má prístup k prehľadom platieb, štatistikám a môže exportovať reporty pre vedenie klubu.
- **Tréner** – prihlasuje svojich zverencov na preteky.
- **Člen klubu** – používateľ, ktorý má prístup k svojmu účtu a prehľadu platieb.

Predpoklady a závislosti

Systém bude nasadený na webhosting s podporou PHP 8 a Laravel frameworku. Funkcionalita párovania platieb bude závislá od prístupu k Fio API a integrácie s Jotform API.

Funkčné požiadavky

FR0: Prihlásenie a roly (používatelia)

FR0.1 Autentifikácia:

- Používateľ sa prihlasuje pomocou email a hesla uloženého v hashovanej forme
- Autentifikácia prebieha cez zabezpečený kanál (HTTPS)
- Pri neúspešných pokusoch o prihlásenie sa vedie log a systém po 5 neúspešných pokusoch účet dočasne zablokuje (napr. na 15 minút)
- Po prihlásení systém overí priradené roly používateľa a nastaví oprávnenia pre prístup k jednotlivým modulom.
- Politika hesiel (min dĺžka, zložitost), rate limiting na prihlásenie, MFA pre admin/manager.

FR0.2 Roly:

Každý používateľ môže mať **jednu alebo viac rolí** (v databáze implementované cez tabuľky `roles` a `user_roles`). Základné roly systému sú:

1. Admin

Správca systému, ktorý má prístup ku všetkým funkciám vrátane správy používateľov, rolí, auditov a systémových nastavení.

Prístupové práva:

- plný prístup k všetkým modulom
- správa používateľov, rolí a oprávnení
- zmena nastavení párovania, notifikácií a systémových parametrov

2. Manažér klubu

Osoba poverená správou členských údajov, predpisov a platieb.

Prístupové práva:

- správa členov, predpisov, transakcií, e-mailov
- manuálne a automatické párovanie
- export reportov a štatistík

3. Tréner (funkcionalita bude zapracovaná ako voliteľná)

Používateľ, ktorý má priradených zverencov (členov).

Prístupové práva:

- prístup iba k svojim zverencom
- môže zverencov prihlasovať na preteky (odošle sa mail s menami a kategóriou, názvom preteku na klubový mail)

4. Člen

Bežný používateľ – člen klubu.

Prístupové práva:

- prístup iba k vlastným predpisom, úhradám a histórii platieb
- možnosť sťahovať potvrdenia o úhrade, generovať QR kód pre platbu (voliteľná funkcionalita)

FR0.3 Správa prístupov

Roly môžu byť priradované a odoberané len používateľom s rolou **Admin** alebo **Manažér klubu**.

Jeden používateľ môže mať viac rolí (napr. „Manažér“ aj „Člen“).

Oprávnenia sa určujú kumulatívne podľa všetkých pridelených rolí.

Zmeny rolí sa zaznamenávajú do **audit logu** (`audit_log`).

FR1: Evidencia členov (clenovia)

FR1.1 Vytvorenie člena:

- **Automaticky** z Jotform API (mapovanie: meno, priezvisko, email, telefón, adresa, dátum narodenia, prípadne údaje zákonného zástupcu).
- **Manuálne** cez UI manažéra.

FR1.2 Unikátnosť: `clenovia.email` a `clenovia.variabilny_symbol` sú **UNIQUE**.

FR1.3 Ochrana osobných údajov: `rodne_cislo` sa ukladá **šifrovane** a je **skryté** v UI (maskovanie; plné zobrazenie len pre manažéra, člena).

FR1.4 Zákonný zástupca: Polia zástupcu sú **NULLABLE** a zobrazia sa len pri **maloletých** (odvodené z `datum_narodenia`).

FR1.5 Stav účtu člena: `stav_uctu` (DECIMAL) je **počítaný**: suma (predpisy) – suma (prijaté platby), a systém ho **udržiava** po každom párovaní/odpárovaní.

FR2 – Používatelia vs. členovia (pouzivatelia ↔ clenovia)

FR2.1 Väzba 1:1: `pouzivatelia.id_clena` odkazuje na `clenovia.id_clena` . Každý **člen** môže mať **jeden** účet používateľa.

FR2.2 Aktivácia/deaktivácia: `pouzivatelia.stav` umožní **deaktivovať** prístup bez zmazania dát.

FR2.3 Registrácia člena: Pri automatickom importe člena môže systém **vytvoriť účet používateľa** (stav aktívny/deaktivovaný podľa nastavení).

FR3 – Predpisy (predpisy)

FR3.1 CRUD:

FR3.1 CRUD: Manažér vytvára/mení/ruší **predpisy** (`id_predpisu` , `id_clena` , `obdobie` , `suma` , `termin_splatnosti` , voliteľné `poznanka`).

FR3.2 VS v predpise: Stĺpec `predpisy.vs` sa predvyplní **podľa** `clenovia.variabilny_symbol` (možno **pre-písať** pre výnimky).

FR3.3 Stav predpisu: `stav` $\in \{\text{nezaplatené, zaplatené, čiastočne, preplatok}\}$ – **udržiava sa automaticky** podľa spárovaných transakcií.

FR3.4 QR pre platbu: Systém generuje **QR payload** (`qr_payload`) a ukladá cestu na PNG (`qr_png_path`) pri vytvorení/zmene predpisu alebo na požiadanie. (voliteľná funkcionálna)

FR3.5 Export: Manažér môže exportovať predpisy (CSV/PDF) **za obdobie** alebo **na člena**. (voliteľná funkcionálna)

FR3.6 Preplatok na predpise: Predpis môže mať stav `preplatok` a hodnotu `preplatok_amount` (≥ 0). V detaile predpisu a v zozname predpisov sa **zvýrazní** informácia „Preplatok: X €“.

FR3.7 Zostatky: Pre každý predpis aplikácia zobrazuje: `predpis_suma` , `uhradene_spolu` , `nedoplatok` , `preplatok` .

FR4 – Import transakcií (transakcie)

FR4.1 Import z Fio API: Mapovanie na schému:

- dátum → `datum` , suma → `objem` , mena → `mena` ,
- protiúčet → `protiucet` , názov protiúčtu → `nazov_protiuctu` , kód/názov banky → `kod_banky` , `nazov_banky` ,

- `ks`, `vs`, `ss`, užívateľská identifikácia → `uzivatelska_identifikacia`, správa pre prijímateľa → `zprava_pre_prijemcu`,
- typ, vykonal (`provedl`), upresnenie, komentár, BIC, `id_pokynu`.

FR4.2 Idempotencia: Primárny kľúč `id_pohybu (BIGINT)` = bankové ID; **opätovný import nesmie duplikovať** záznam.

Garantovať idempotenciu importu Fio cez PRIMARY KEY `id_pohybu` a kontrolu deduplikácie na aplikačnej úrovni (retry safety).

FR4.3 Pseudonymizácia v exportoch: Exporty pre členov **nezobrazujú celé čísla protiúčtov** (maskovať). (voliteľná funkcionálna) GDPR?

FR5 - Automatické párovanie (transakcie ↔ predpisy)

FR5.1 Poradie pravidiel:

1. **VS** (`transakcie.vs` = `predpisy.vs`) a **zhoda sumy** → priama zhoda.
2. Ak VS chýba: podľa **protiúčtu** známeho pre daného člena (z histórie úspešne spárovaných platieb) + zhoda sumy.
3. Ak účet nepoznáme: podľa **mena**/poznámky (normalizácia bez diakritiky, medzier, veľkosti písma) + zhoda sumy.

FR5.2 Skóre a nejednoznačnosť: Pravidlá pridelujú skóre (VS=100, účet=80, meno/poznámka=60). Ak rozdiel < 20 bodov → **nejednoznačné**, nepáruj automaticky.

FR5.3 Čiastočné úhrady a preplatky:

- **Čiastočné:** priradiť, ponechaj zvyšok otvorený.
- **Zachytenie preplatku na danom predpise (lokálne):** Ak úhrada > `predpis_suma`, rozdiel sa **pripíše ako preplatok** na tento predpis (nepresúva sa na iné predpisy).
- **Alternatívna politika (globálne):** V nastaveniach je prepínač:
 - „Preplatok ponechať na danom predpise“
 - „Preplatok alokovať na najstarší otvorený predpis člena“

FR5.4 Audit: Pri vzniku preplatku sa zaloguje: predpis, transakcia, výška preplatku, pravidlo (lokálne/globálne).

FR6 - Manuálne párovanie

FR6.1 UI návrhov: Manažér vidí **zoradené návrhy** podľa skóre; vie priradiť **jednu transakciu k jednému predpisu** alebo **rozdeliť** (ak systém umožní čiastočné párovanie).

FR6.2 Odpárovanie: Manažér môže **odpárovať** transakciu; systém upraví `predpisy.stav`, **prepočíta** `clenovia.stav_uctu` a zaznamená audit.

FR6.3 Ignorovať: Transakciu možno označiť ako **ignorovať** (napr. dar, omyl).

FR7 – E-mailové notifikácie (emaily)

FR7.1 Odosielanie: Pri udalostiach (nový predpis, nedoplatok) systém odošle e-mail; uloží záznam do `emaily` (`id_clena`, `predmet`, `obsah`, `datum_odoslania`, `stav`, `prilohy`).

FR7.2 Retry: Zlyhané (`stav='zlyhalo'`) sa pokúsime **znovu odoslať**; udržiavať počet pokusov v internom logu.

FR7.3 Šablóny: Obsah generovaný zo šablón; prílohy môžu zahŕňať **QR**.

FR8 – Reporty a prehľady

FR8.1 Manažér:

- Pohľad **dlžníci podľa obdobia**, nedoplatky/preplatky, **export CSV**.
- Prehľad podľa člena: predpisy, úhrady, **stav_uctu**.

FR8.2 Člen:

- Prehľad **vlastných** predpisov a úhrad;
- **stiahnutie potvrdenia** pre uplatnenie príspevku uznanému športu (voliteľná funkcionálna).

FR9 – Integrita a pravidlá

FR9.1 Zmazanie člena: Nie je povolené, ak má **existujúce neuhradené predpisy alebo nepriradené transakcie**.

FR9.2 Zmazanie predpisu: Zakázané, ak je **spárovaný** (povolené zrušiť len po **odpárovaní**).

FR9.3 Konzistencia: Spárovanie **musí** aktualizovať `predpisy.stav` a **rekalkulovať** `clenovia.stav_uctu`.

FR10 – Audit (naprieč modulmi)

FR10.1 Audit trail: Zmeny kľúčových entít (clenovia, predpisy, manuálne párovanie) sa **logujú**: kto, kedy, čo (pred/po).

FR10.2 Prístupy k citlivým dátam: Každý **plný náhľad** na `rodne_cislo` sa zaznamená (právne odôvodnenie). ???

FR11 – Nastavenia systému

FR11.1 Párovanie: Prepínače: povoliť **čiasťočné** párovanie, politika **preplatkov**, prah **skóre** pre auto-párovanie.

FR11.2 Predpisy: Default `obdobie` (napr. „YYYY/YY“), default **VS** = `variabilny_symbol` člena, formátovanie VS.

FR11.3 E-mail: Povoliť/zakázať typy notifikácií; odosielateľ; DKIM/DMARC nastavenia (mimo ERD, ale systémová požiadavka).

Nefunkčné požiadavky

NFR1: Systém musí byť dostupný 24/7 s plánovanými prestávkami na údržbu.

NFR2: Odozva systému nesmie presiahnuť 3 sekundy pri bežnom zaťažení.

NFR3: Dáta používateľov musia byť uložené bezpečne (hashované heslá, HTTPS pripojenie).

NFR4: Používateľské rozhranie musí byť prehľadné, responzívne a prístupné z mobilných zariadení.

NFR5: Systém musí byť navrhnutý modulárne pre jednoduchú údržbu a rozšírenie.

Prílohy