**Technisch Ontwerp Slim Minder MVP**

**Inleiding**

Slim Minder is een **sociaal-fintech platform** gericht op schuldpreventie en gedragsverandering in de persoonlijke financiën. De app combineert **online bankieren** met **budgetcoaching**: gebruikers koppelen hun bankrekening(en) via PSD2 en krijgen direct inzicht in inkomsten en uitgaven[binnenlandsbestuur.nl](https://www.binnenlandsbestuur.nl/sociaal/lokale-lasten-ozb/buddy-app-wil-schulden-voorkomen#:~:text=De%20app%20is%20een%20vorm,wordt%20hulpverlening%20laagdrempelig%20en%20schaalbaar). Met Slim Minder kunnen gebruikers bijvoorbeeld hun **vaste lasten apart zetten, automatisch sparen,** en gepersonaliseerde **financiële tips** ontvangen[binnenlandsbestuur.nl](https://www.binnenlandsbestuur.nl/sociaal/lokale-lasten-ozb/buddy-app-wil-schulden-voorkomen#:~:text=De%20app%20is%20een%20vorm,wordt%20hulpverlening%20laagdrempelig%20en%20schaalbaar). Ook checkt de app automatisch of de gebruiker in aanmerking komt voor relevante **toeslagen of lokale inkomensregelingen**[binnenlandsbestuur.nl](https://www.binnenlandsbestuur.nl/sociaal/lokale-lasten-ozb/buddy-app-wil-schulden-voorkomen#:~:text=De%20app%20is%20een%20vorm,wordt%20hulpverlening%20laagdrempelig%20en%20schaalbaar). Dit alles moet voorkomen dat kleine geldzorgen uitgroeien tot problematische schulden. In Nederland hebben immers ~2 miljoen huishoudens al schulden of lopen risico daarop[binnenlandsbestuur.nl](https://www.binnenlandsbestuur.nl/sociaal/lokale-lasten-ozb/buddy-app-wil-schulden-voorkomen#:~:text=De%20app%20is%20een%20vorm,wordt%20hulpverlening%20laagdrempelig%20en%20schaalbaar)[binnenlandsbestuur.nl](https://www.binnenlandsbestuur.nl/sociaal/lokale-lasten-ozb/buddy-app-wil-schulden-voorkomen#:~:text=%27Nederland%20kent%202%20miljoen%20huishoudens,wordt%20hulpverlening%20laagdrempelig%20en%20schaalbaar), wat de samenleving jaarlijks circa **€17 miljard** kost[binnenlandsbestuur.nl](https://www.binnenlandsbestuur.nl/sociaal/lokale-lasten-ozb/buddy-app-wil-schulden-voorkomen#:~:text=de%20effecten%20van%20de%20app,besparen%2C%20wordt%20ook%20nog%20onderzocht). Slim Minder wil deze groep ondersteunen met laagdrempelige, schaalbare hulpverlening[binnenlandsbestuur.nl](https://www.binnenlandsbestuur.nl/sociaal/lokale-lasten-ozb/buddy-app-wil-schulden-voorkomen#:~:text=%27Nederland%20kent%202%20miljoen%20huishoudens,wordt%20hulpverlening%20laagdrempelig%20en%20schaalbaar), zodat financiële problemen tijdig worden aangepakt.

De kern van Slim Minder is gedragsverandering door **inzicht, nudging en gamification**. Gebruikers krijgen **inzichtelijke overzichten** van hun bestedingsgedrag, en de app geeft **nudges** – bijvoorbeeld waarschuwingen bij ongewoon hoge uitgaven of vriendelijke herinneringen om binnen budget te blijven. Tevens bevat Slim Minder **gamification**-elementen: sparen en budgetteren wordt bijna een spel, met **badges, punten en uitdagingen** als beloning voor positief financieel gedrag. Zulke spelelementen en notificaties bevorderen namelijk het gevoel van competentie en autonomie bij gebruikers en motiveren hen om de app consistent te gebruiken[selfdeterminationtheory.org](https://selfdeterminationtheory.org/wp-content/uploads/2024/03/2021_BitrianBuilCatalan_IJBM.pdf#:~:text=%28e,apps%20designed%20to%20manage%20the). Uiteindelijk beoogt Slim Minder bij te dragen aan duurzame financiële zelfredzaamheid: **“meer grip op geld”** voor jong en oud, zodat schulden worden voorkomen voordat ze ontstaan.

De MVP (Minimum Viable Product) van Slim Minder zal vóór eind 2025 worden ontwikkeld voor **iOS**, **Android** en **web**. Hiervoor is gekozen voor één technologie-stack gebaseerd op **React**: we gebruiken **React Native** voor de mobiele apps en **React** voor de webapplicatie, zodat code en kennis herbruikbaar zijn over de platformen heen. Dit technisch ontwerpdocument beschrijft de volledige scope en architectuur van de MVP. Het is bedoeld als praktische leidraad voor het development-team en behandelt gebruikersverhalen, functionele en niet-functionele eisen, de beoogde architectuur, roadmap en testaanpak.

**Gebruikersverhalen (User Stories)**

Om de requirements tastbaar te maken, beschrijven we hier **persona-gebaseerde gebruikersverhalen**. Deze verhalen laten zien hoe verschillende typen gebruikers Slim Minder zouden gebruiken, inclusief randgevallen zoals beperkte bankkoppeling of lage digitale vaardigheden.

* **Jongere (18-25 jaar)** – *Als beginnende jongvolwassene* wil ik overzicht houden op mijn uitgaven zodra mijn salaris binnenkomt. **Scenario:** Sarah (22) koppelt haar betaalrekening en ziet direct in Slim Minder waar haar geld heen gaat. Ze stelt **budgetten** in voor kleding en uitgaan. Wanneer ze deze maand haar kledingbudget bijna overschrijdt, krijgt ze een **nudge**: een pushmelding met “Let op: je zit op 90% van je kledingbudget”. Hierdoor past ze haar uitgaven aan. Ook spaart Sarah voor een nieuwe telefoon; Slim Minder raadt haar een passend **spaardoel** aan en motiveert haar met **badges** als ze elke week iets opzij zet. *Edge-case:* Sarah’s bank ondersteunt PSD2, maar ze heeft geen creditcard – Slim Minder houdt hier rekening mee door alleen haar betaalrekening te monitoren.
* **Gezinshoofd (30-45 jaar)** – *Als vader/moeder van een gezin* wil ik op één plek onze gezamenlijke financiën beheren en ongewenste verrassingen voorkomen. **Scenario:** David (37) koppelt zowel zijn als de rekening van zijn partner in Slim Minder. De app **categoriseert automatisch** alle transacties (salaris, boodschappen, hypotheek, enz.) zodat David ziet waar het gezinsbudget naartoe gaat. Hij stelt voor elke categorie maandelijkse limieten in. Bij overschrijding van bijvoorbeeld het boodschappengeld krijgt hij een **waarschuwing** en tips om de kosten te drukken (bijv. “Je hebt deze maand €50 meer aan boodschappen uitgegeven dan begroot – bekijk onze bespaartips”). David gebruikt het **dashboard** om zijn maandrapport te bekijken en deelt dit met zijn partner, zodat ze samen geïnformeerde beslissingen nemen. *Edge-case:* David’s partner wil haar bank niet koppelen uit privacy-zorgen – Slim Minder biedt daarom de optie om handmatig bedragen toe te voegen, zodat toch een globaal overzicht ontstaat.
* **Senior (65+ jaar)** – *Als gepensioneerde* wil ik eenvoudig kunnen zien of ik binnen mijn pensioeninkomen blijf en of ik in aanmerking kom voor toeslagen. **Scenario:** Mevrouw Jansen (70) gebruikt Slim Minder op haar iPad. Dankzij de **eenvoudige interface** (grote letters, duidelijke iconen) navigeert ze moeiteloos. Ze koppelt haar bankrekening en Slim Minder toont met **grafieken** haar maandelijkse uitgaven aan zorg, huishouden, etc. Wanneer haar uitgavenpatroon verandert, krijgt ze een melding (“U geeft deze maand meer uit aan medicijnen dan gebruikelijk”). Slim Minder helpt haar ook door te **controleren op toeslagen**; de app laat weten dat ze mogelijk recht heeft op zorgtoeslag en legt in begrijpelijke taal uit hoe ze dit aanvraagt. *Edge-case:* Mevrouw Jansen heeft lage digitale vaardigheden – Slim Minder begeleidt haar stap-voor-stap bij de eerste installatie (met een **tutorialmodus**) en biedt desgewenst telefonische ondersteuning via een helpdesk.

**Randgevallen en speciale behoeften:** Naast de drie hoofdpersona’s houdt Slim Minder rekening met diverse situaties. Bijvoorbeeld een **gebruiker zonder bankkoppeling** (bijv. bewust bankvrij of iemand met alleen contant geld) kan de app in beperkte modus gebruiken door zelf transacties of kasstromen in te voeren. Gebruikers met **lage digitale vaardigheden** of **beperkingen** profiteren van onze focus op **toegankelijkheid**: de app is zó ontworpen dat ook minder technisch onderlegde mensen, laaggeletterden of visueel beperkte gebruikers ermee overweg kunnen (heldere taal, voorleesopties, contrastvolle weergave, etc.). Verder is er aandacht voor multi-account gebruikers (bijv. iemand met meerdere bankrekeningen bij verschillende banken) – Slim Minder consolideert deze gegevens tot één overzicht. Door rekening te houden met zulke edge-cases zorgen we dat de MVP bruikbaar is voor een breed en inclusief publiek.

**Functionele specificaties**

Onderstaande lijst beschrijft de **kernfunctionaliteiten** van de Slim Minder MVP. Elke functie is essentieel om gebruikers inzicht te geven in hun financiën, hen tijdig bij te sturen (nudging) en positief financieel gedrag te belonen (gamification).

**Bankkoppeling via PSD2 (Open Banking)**

Slim Minder maakt gebruik van **PSD2** (Revised Payment Services Directive) om veilig verbinding te maken met bankrekeningen van de gebruiker. Via een **OAuth2-gebaseerde** koppeling geeft de gebruiker expliciet toestemming aan de app om transactiedata uit te lezen[appclonescript.com](https://www.appclonescript.com/securing-securing-fintech-apps-with-encryption-oauth-2-0-mfawith-encryption-oauth-2-0/#:~:text=,as%20the%20EU%20and%20UK). We integreren een betrouwbare **PSD2-partner** (zoals **Klarna Kosma**, **Invers** of **Salt Edge**) die een Access-to-Account (XS2A) API biedt naar de meeste Nederlandse banken. Dit betekent dat de gebruiker met een paar klikken zijn bank kan verbinden en dat wij via één gestandaardiseerde API toegang krijgen tot rekeninginformatie van duizenden banken[saltedge.com](https://www.saltedge.com/products/data_enrichment#:~:text=The%20Data%20Enrichment%20platform%20can,on%20automatic%20account%20statement%20gathering). De koppeling voldoet aan strenge veiligheidsnormen (bankiers weten wie wij zijn als TPP, sterke klantauthenticatie/SCA wordt toegepast). Het resultaat is dat Slim Minder automatisch de transacties, saldo en rekeninggegevens kan binnenhalen, zonder dat de gebruiker handmatig bankafschriften hoeft te importeren. **Storing/Timeout scenario’s:** de app houdt rekening met scenario’s waarin bankkoppelingen tijdelijk niet werken (bijv. PSD2 API down); in zulke gevallen tonen we een foutmelding en proberen we het later opnieuw, en de gebruiker behoudt toegang tot laatste bekende data offline.

**Automatische categorisatie van transacties**

Eenmaal opgehaald, worden alle **banktransacties automatisch gecategoriseerd**. Slim Minder gebruikt hiervoor data-enrichmentsservices van de PSD2-partner of eigen logica. Bijvoorbeeld, een PIN-transactie bij “AH To Go” wordt herkend als **categorie “Boodschappen”**. Dankzij machine learning-algoritmes kan de categorisatie-engine ook minder duidelijke transacties een categorie toewijzen en zelfs nieuwe inkomsten herkennen[saltedge.com](https://www.saltedge.com/products/data_enrichment#:~:text=The%20categorisation%20API%20assigns%20a,by%20companies%20and%20their%20customers). We ondersteunen standaard categorieën (Wonen, Vervoer, Voeding, Vrije Tijd, etc.) en laten de gebruiker eventueel categorieën aanpassen of eigen labels toevoegen voor persoonlijke controle. Deze automatische categorisatie bespaart gebruikers veel tijd en levert **“meaningful actionable insights”** op vanuit ruwe transactiegegevens[saltedge.com](https://www.saltedge.com/products/data_enrichment#:~:text=The%20categorisation%20API%20assigns%20a,by%20companies%20and%20their%20customers). **Edge-case:** Mocht een transactie echt onbekend zijn, dan plaatst Slim Minder deze in “Overig” en kan de gebruiker zelf een categorie kiezen; het systeem leert hiervan voor de toekomst (self-learning).

**Inzichtelijke grafieken en budgetten per categorie**

Om **inzicht** te bevorderen, biedt Slim Minder een rijk **dashboard** met grafieken, overzichten en budgetteringstools. Gebruikers zien in één oogopslag hun **inkomsten vs. uitgaven** per maand, opgesplitst naar categorie. **Datavisualisatie** is hier key: we tonen interactieve cirkeldiagrammen (waaraan zie je welke categorieën het meest uitgeven), staafdiagrammen voor uitgaven per maand, en trendlijnen om bv. spaargeldgroei te tonen. De gebruiker kan per uitgaven-categorie **budgetten instellen** (bijv. €300 per maand voor boodschappen, €50 voor koffie). Een voortgangsbalk of grafiek laat vervolgens real-time de status zien: hoeveel van het budget is al gebruikt en hoeveel is nog over. Dit **real-time bijhouden en visueel presenteren** van de voortgang helpt gebruikers om te zien hoe ze het doen ten opzichte van hun doelen, wat het gevoel van controle en competentie vergroot[selfdeterminationtheory.org](https://selfdeterminationtheory.org/wp-content/uploads/2024/03/2021_BitrianBuilCatalan_IJBM.pdf#:~:text=users%20to%20achieve%20their%20financial,app%20alerts%20and). Via intuïtieve kleurcodes (groen/oranje/rood) ziet men of men binnen budget blijft. Ook kunnen gebruikers op categorieën tikken voor details of om verleden maanden te vergelijken – zo wordt inzicht geboden in **uitgavenpatronen over de tijd**.

**Waarschuwingen en nudges bij overschrijdingen**

Slim Minder fungeert als een **virtuele coach** die op de achtergrond meekijkt en *proactief* ingrijpt bij afwijkingen. Zodra een gebruiker op weg is een budget te overschrijden of ongebruikelijk veel uitgeeft binnen korte tijd, genereert de app een **waarschuwing** of *nudge*. Voorbeelden: *“Je hebt al 90% van je restaurantbudget verbruikt, denk eraan dat je jezelf een limiet had gesteld”* of *“Let op: je geeft deze week 20% meer uit dan gemiddeld”*. Deze meldingen komen als pushbericht op de telefoon en in-app notificaties. Het doel is gebruikers **bewust te maken** van hun gedrag en hen subtiel aan te sporen tot bijsturing voordat er echt financiële problemen ontstaan. De frequentie en toon van nudges zijn zorgvuldig gekozen (niet te belerend, wel duidelijk). Naast waarschuwingen bij overschrijding zijn er ook **herinneringen** (bijv. *“Je spaardoel ‘Nieuwe laptop’ is nog €1000 weg – overweeg deze maand iets extra opzij te zetten”*). Al deze nudges zijn gebaseerd op inzichten uit gedragspsychologie dat tijdige feedback en positieve bekrachtiging effectief zijn voor gedragsverandering[selfdeterminationtheory.org](https://selfdeterminationtheory.org/wp-content/uploads/2024/03/2021_BitrianBuilCatalan_IJBM.pdf#:~:text=%28e,apps%20designed%20to%20manage%20the). **Edge-case:** Gebruikers kunnen instellen welke meldingen ze willen ontvangen (iemand met heel onregelmatig inkomen wil misschien minder frequentie aan waarschuwingen om stress te voorkomen). Ook worden notificaties gepauzeerd als de gebruiker bijvoorbeeld op vakantie is (handmatig in te stellen) om irrelevante nudges te vermijden.

**Bespaardoelen en voortgang**

Naast budgetteren voor uitgaven, richt Slim Minder zich ook op het **positieve aspect: sparen**. Gebruikers kunnen **bespaardoelen** instellen, bijvoorbeeld *“€500 noodfonds”* of *“€1500 voor vakantie”*. De app helpt bij het kiezen van realistische doelen (gebaseerd op het inkomen en de uitgavenhistorie) en verdeelt grotere doelen in **haalbare maandelijkse subdoelen**. Iedere keer dat de gebruiker geld opzij zet richting een doel, kan dit in de app geregistreerd worden (of automatisch gedetecteerd worden als er een overboeking naar een spaarrekening plaatsvindt). Slim Minder toont de **voortgang** met progressiebalken en percentielen – bijvoorbeeld *“60% van je ‘Vakantie’ doel bereikt”*. Deze visuele feedback houdt sparen leuk en **tastbaar**. Wanneer een doel is bereikt, feliciteert de app de gebruiker en kan eventueel voorstellen om een nieuw doel aan te maken, waarmee de cyclus van financieel positieve acties doorloopt. Ook deelt Slim Minder tips om sneller te sparen (bijv. *“Als je €20 per week bespaart op uiteten, haal je je doel 1 maand eerder”*). Dit koppelt de bespaar- en budgetfunctionaliteit aan elkaar om gebruikers te stimuleren bewuste keuzes te maken.

**Gamification: badges, challenges en beloningen**

Om de betrokkenheid hoog te houden, bevat de MVP diverse **gamification**-elementen. Gebruikers kunnen **badges** verdienen voor goed gedrag, bijvoorbeeld *“Eerste spaardoel behaald”*, *“3 maanden op rij binnen budget”* of *“Uitgaven 10% verlaagd tov vorige maand”*. Deze badges verschijnen in hun profiel en geven een gevoel van prestatie. Daarnaast kunnen we **challenges** introduceren, zoals een 30-dagen bespaarchallenge (waarbij de gebruiker elke dag een kleine tip krijgt en progressie boekt) of een uitdaging om een week lang niet te bestellen bij dure bezorgdiensten. Het behalen van challenges levert weer beloningen op in de app – denk aan extra badges of een hogere **“Slim Rank”** (een fictieve score die aangeeft hoe bewust iemand met geld omgaat). Hoewel er geen materiële beloning is, zorgen deze spelelementen voor **competentiegevoel en autonomie**: de gebruiker ziet concreet resultaat van zijn inspanningen en heeft deels vrijheid in welke uitdagingen hij aangaat[selfdeterminationtheory.org](https://selfdeterminationtheory.org/wp-content/uploads/2024/03/2021_BitrianBuilCatalan_IJBM.pdf#:~:text=%28e,apps%20designed%20to%20manage%20the). Gamification maakt het managen van geld minder een plicht en meer een *motiverend proces*. Let wel: de toon blijft volwassen en respectvol – het is geen kinderlijk spel, maar subtiele speelse elementen geïntegreerd in een serieuze financieel hulpmiddel. **Edge-case:** Voor gebruikers die hier geen behoefte aan hebben, is gamification optioneel; men kan badges/challenges negeren of eventueel de weergave ervan uitschakelen.

**Persoonlijke AI-coach (chat interface met NLP)**

Een unieke feature van Slim Minder is de **persoonlijke AI-coach**: een intelligente chatbotsysteem dat de gebruiker te woord staat voor financiële vragen en coaching. In de MVP zal deze coach een **chat-interface** hebben (vergelijkbaar met chatapps) waar de gebruiker bijvoorbeeld kan typen: “Hoe kan ik deze maand €50 extra besparen?” of “Waarom was mijn energierekening hoger dan normaal?”. De AI-coach, mogelijk aangedreven door een NLP-model zoals GPT-4 of een gespecialiseerde Nederlandse taalmodel, begrijpt de vraag en geeft daarop advies. Bijvoorbeeld: *“Je kunt overwegen om je abonnement op streamingdiensten te pauzeren, dat bespaart ~€15 p.m. Verder zie ik dat je veel uitgeeft aan uit eten; misschien kun je deze week een keer extra thuis koken?”*. De AI heeft toegang (read-only) tot de categorie-overzichten van de gebruiker om contextueel relevante tips te geven. Daarnaast kan de coach algemene vragen beantwoorden (*“Wat betekent rente op rente?”*), zodat gebruikers ook hun financiële kennis vergroten. Belangrijk is dat de AI **conversatie in gewone mensentaal** voert, zodat ook minder financieel onderlegde gebruikers advies begrijpen. Voor de implementatie integreren we een cloud-gebaseerde NLP-service (bijv. **OpenAI API** voor geavanceerde conversatie, omdat die meerdere talen inclusief Nederlands ondersteunt, of een alternatief als er specifieke eisen zijn qua data-opslag). We nemen maatregelen voor **AI-output validatie** (zie testplan) om te zorgen dat adviezen correct en veilig zijn. In latere versies kan deze coach zich personaliseren (leren van voorkeuren) en proactief tips geven, maar in de MVP reageert hij vooral op vragen en eenvoudige triggers. **Edge-case:** Als de AI een vraag niet kan beantwoorden (bijv. te complex of buiten scope), zal hij dit netjes aangeven en eventueel doorverwijzen naar een hulppagina of menselijk adviseur.

**Dashboard met maandoverzicht en rapportage**

Elke gebruiker heeft toegang tot een **persoonlijk dashboard** dat een holistisch overzicht van de financiën biedt. Dit dashboard fungeert als **startscherm** in de app en op de webversie. Hierop staan bijvoorbeeld: **samenvatting van de maand** (totale inkomsten, totale uitgaven, saldo-opbouw), top-3 categorieën waar geld naartoe ging, en een indicator of men onder/over budget zit. Daarnaast biedt het dashboard toegang tot een **rapportagefunctie**: gebruikers kunnen per maand (of andere periode) een rapport genereren/exporteren met alle relevante cijfers, handig voor bijvoorbeeld persoonlijke administratie of om te bespreken met een budgetcoach. Dit rapport kan worden weergegeven in de app als infografiek, maar ook bijvoorbeeld als **PDF-download** voor archivering. Voor de MVP zorgen we dat het rapport de belangrijkste onderdelen dekt: inkomsten vs. uitgaven per categorie, lijst van overschrijdingen/nudges die maand, spaargeld-evolutie, etc. Het dashboard en rapportage helpen gebruikers reflecteren op hun gedrag: *“Heb ik deze maand progressie gemaakt?”* of *“Waar ging het mis en waar ging het goed?”*. Door deze inzichten toegankelijk te maken, bevorderen we continu leren en verbeteren. **Edge-case:** Gebruikers zonder ervaring met financiële overzichten worden niet overladen met data – we bieden een **“samenvatting in plain language”** modus, bijvoorbeeld een korte tekst: *“In september heeft u €100 meer gespaard dan in augustus, goed gedaan!”* naast de cijfers. Dit maakt de rapportage ook voor leken begrijpelijk.

**Accountsysteem met Free/Core/Premium verschillen**

Slim Minder hanteert een **accountsysteem met verschillende abonnementsniveaus** om functionaliteit stapsgewijs beschikbaar te maken en opschaalbaarheid van de dienst te waarborgen. In de MVP onderscheiden we drie niveaus: **Free, Core (Standaard)** en **Premium**:

* **Free (Gratis):** Nieuwe gebruikers kunnen gratis aan de slag met basisfunctionaliteit. Dit omvat doorgaans het koppelen van één bankrekening, basis categorie-overzichten, en een beperkt aantal budgetten of spaardoelen. Ook kunnen ze een paar algemene tips krijgen en een voorproef van de AI-coach (bijv. 5 vragen per maand gratis). Free-gebruikers zien mogelijk lichte **in-app promoties** om te upgraden, maar geen storende advertenties van derden (we bewaren de gebruikerservaring).
* **Core (Standaard):** Dit is het kernaanbod (mogelijk tegen een kleine maandelijkse vergoeding of gesponsord via partners, afhankelijk van het verdienmodel). Core-leden krijgen toegang tot alle **hoofdfuncties** van de MVP: onbeperkt bankrekeningen koppelen, volledige transactiecategorieën, onbeperkt budgetten/spaardoelen instellen, waarschuwingen/nudges op maat, en gebruik van de AI-coach zonder limiet. De meeste beschouwde functionaliteiten vallen in deze categorie, zodat de app effectief is voor schuldpreventie.
* **Premium:** Voor gebruikers die extra verdieping willen, biedt Premium enkele **advanced features** bovenop Core. Denk aan **proactieve coaching** (de AI-coach komt zelf met analyses, of er is toegang tot een menselijke financieel coach via chat), meer uitgebreide rapportages (bijv. kwartaal- of jaaranalyse, of export naar Excel), **gezinsaccounts** (meerdere gebruikers in één huishouden kunnen data delen in de app), en diepere **personalisatie** (zoals eigen categorieën, of integratie met andere diensten als kalender voor rekeningafspraken). In de MVP fase zullen Premium-features deels conceptueel aanwezig zijn om het onderscheid duidelijk te maken, maar de volledige implementatie kan in latere versies plaatsvinden. Wel zal het accountsysteem in de MVP technisch gereed zijn om onderscheid te maken: gebruikers registreren zich met e-mail of via federated login, en de server kent rechten toe op basis van hun plan.

**Account- en gebruikersbeheer:** Alle gebruikers (Free/Core/Premium) hebben een beveiligd account (e-mail + wachtwoord of OAuth login). We implementeren **e-mailverificatie** bij registratie en ondersteunen optioneel **2FA** (twee-factor-authenticatie) voor extra beveiliging bij het inloggen, vooral relevant voor Premium gebruikers met meer gevoelige data. Upgraden/downgraden tussen plannen gebeurt soepel in-app. Dit accountsysteem zorgt tevens dat gebruikersdata gescheiden blijft (multitenancy op de backend) en dat bijvoorbeeld Premium-features makkelijk aan- of uit te zetten zijn per account.

**Niet-functionele eisen**

Naast de functionele requirements zijn er belangrijke **niet-functionele eisen** waaraan Slim Minder moet voldoen. Deze hebben betrekking op beveiliging, prestaties, privacy, schaalbaarheid, toegankelijkheid en taal. Hieronder lichten we deze aspecten toe.

**Security (Beveiliging)**

Gezien de sensitieve financiële data die Slim Minder verwerkt, staat **beveiliging** bovenaan. We hanteren **bank-level security** standaarden om het vertrouwen van gebruikers en compliance te waarborgen. Enkele concrete maatregelen:

* **Authenticatie & Autorisatie:** We gebruiken **OAuth 2.0** protocollen voor de bankkoppeling en voor eventuele integraties met externe diensten. Dit zorgt ervoor dat gebruikersdata van banken gedeeld kan worden zonder dat Slim Minder ooit de inloggegevens van de gebruiker hoeft te kennen[appclonescript.com](https://www.appclonescript.com/securing-securing-fintech-apps-with-encryption-oauth-2-0-mfawith-encryption-oauth-2-0/#:~:text=,as%20the%20EU%20and%20UK). Voor onze eigen login-procedures implementeren we veilige wachtwoordopslag (bcrypt hashing) en bieden we optionele **Multi-Factor Authentication (MFA)** zoals een SMS of authenticator app-code bij inloggen. Dit voegt een extra beveiligingslaag toe naast gebruikersnaam/wachtwoord, wat met name cruciaal is voor financiële applicaties[appclonescript.com](https://www.appclonescript.com/securing-securing-fintech-apps-with-encryption-oauth-2-0-mfawith-encryption-oauth-2-0/#:~:text=Multi,Verification)[appclonescript.com](https://www.appclonescript.com/securing-securing-fintech-apps-with-encryption-oauth-2-0-mfawith-encryption-oauth-2-0/#:~:text=,reduces%20fraud%20and%20identity%20theft).
* **Encryptie:** Alle dataverkeer is versleuteld via **HTTPS/TLS 1.3**. Gevoelige persoonsgegevens en bankgegevens die op de server terechtkomen, worden **end-to-end encrypted** waar mogelijk – d.w.z. versleuteld bij de bron (device) en pas ontsleuteld als het echt nodig is voor verwerking[appclonescript.com](https://www.appclonescript.com/securing-securing-fintech-apps-with-encryption-oauth-2-0-mfawith-encryption-oauth-2-0/#:~:text=End,using%20reliable%20banking%20encryption%20software)[appclonescript.com](https://www.appclonescript.com/securing-securing-fintech-apps-with-encryption-oauth-2-0-mfawith-encryption-oauth-2-0/#:~:text=By%20combining%20E2EE%2C%20OAuth%202,strong%20defense%20against%20evolving%20cyber). In de database worden kritieke velden (bijv. tokens voor banktoegang, persoonlijke identificatie) versleuteld opgeslagen (AES-256). Zo is de data zelfs bij een datalek onleesbaar voor derden.
* **Autorisatie & Least Privilege:** Binnen de architectuur zorgen we voor strikte **toegangsrechten**. Bijvoorbeeld, de mobiele app en webapp communiceren met de backend via een token die slechts toegang geeft tot de eigen data van die gebruiker (JWT met scopes). Services binnen de backend krijgen alleen toegang tot data die ze nodig hebben (microservices met aparte rollen).
* **GDPR compliance:** Slim Minder voldoet aan de AVG (GDPR). Dat betekent: heldere gebruikersconsent voor datagebruik, recht van inzage en verwijdering van data, en **dataminimalisatie** (we verzamelen niet meer dan noodzakelijk). Persoonsdata wordt opgeslagen op servers binnen de EU. Een Privacy Impact Assessment wordt gedaan om risico’s in kaart te brengen en te mitigeren.
* **Veilige ontwikkeling en testing:** Tijdens development volgen we OWASP-richtlijnen om veelvoorkomende beveiligingslekken (SQL-injectie, XSS, CSRF, etc.) te vermijden. We plannen **penetratietesten** en code-audits in de MVP-fase om kwetsbaarheden vroeg op te sporen.

Kortom, door **E2EE, OAuth2.0 en MFA te combineren creëren we een sterk meerlaagse beveiligingsomgeving**[**appclonescript.com**](https://www.appclonescript.com/securing-securing-fintech-apps-with-encryption-oauth-2-0-mfawith-encryption-oauth-2-0/#:~:text=By%20combining%20E2EE%2C%20OAuth%202,strong%20defense%20against%20evolving%20cyber), waarbij gebruikersvertrouwen en naleving van regelgeving centraal staan.

**Performantie (real-time en responsiveness)**

Slim Minder moet **snel en responsief** aanvoelen, ondanks de hoeveelheid data (transacties) en analyses. Belangrijke performance-eisen:

* **Real-time waarschuwingen:** Wanneer er een nieuwe transactie binnenkomt (via de PSD2-webhook of periodieke polling), moet Slim Minder binnen enkele minuten de data verwerken en relevante nudges versturen. De gebruiker ervaart een nagenoeg **real-time update** van zijn saldo en budgetten. Dit betekent dat onze backend efficiënt om moet gaan met inkomende bankfeeds en de logica voor waarschuwingen asynchroon maar snel moet draaien (bijv. via event-driven architecture of job queue die direct nudges uitstuurt).
* **App snelheid:** De mobiele app en webapp moeten soepel scrollen door transactie-overzichten, grafieken snel renderen, en chat-antwoorden van de AI-coach binnen enkele seconden tonen. We optimaliseren de **frontend** door gebruik van React’s efficiënte rendering, lokale caching (bijv. recent opgehaalde data opslaan op het toestel voor snelle toegang) en lazy loading van minder kritieke onderdelen. Animaties en grafieken worden geoptimaliseerd (eventueel via canvas of chart-libraries die performance hoog houden).
* **API-responsiviteit:** De backend-API die de frontend aanstuurt, moet binnen **<200ms** responderen op de meeste calls (exclusief externe PSD2 calls). We bereiken dit door caching van veelgebruikte data (bijv. categorie-samenvattingen die één keer per dag berekend worden), indexering in de database voor snelle queries en eventueel gebruik van een in-memory cache (Redis) voor sessiegegevens en veelgevraagde resources. Als bijvoorbeeld de gebruiker zijn dashboard opent, komt een groot deel van de data uit eerder berekende aggregaties i.p.v. op dat moment zware berekeningen uit te voeren.
* **Schalen onder load:** Hoewel MVP gebruikersaantallen beperkt zijn, ontwerpen we voor voren mee dat de app hoge gelijktijdige load aan kan. De cloud-infrastructuur (zie onder) kan **autoscalen** bij piekverkeer, en we testen de app voor scenario’s als: 1000 gelijktijdige gebruikers die een notificatie ontvangen of de AI-coach bevraagt. Performantie wordt voortdurend gemonitord zodat we regressies meteen zien.
* **Edge-case netwerk:** In mobiele context kan de verbinding traag of offline zijn. De app biedt een *offline modus* voor de laatste bekende gegevens: d.w.z. als je even geen internet hebt, kun je nog je laatst bijgewerkte dashboard zien. Synchronisatie gebeurt weer op de achtergrond zodra er verbinding is. Dit zorgt voor een vloeiende gebruikerservaring zelfs bij matige connectie.

**Privacy (minimale dataopslag & gebruikerscontrole)**

Privacy is nauw verweven met beveiliging, maar richt zich op **dataminimalisatie en controle** voor de gebruiker:

* **Minimale dataopslag:** Slim Minder slaat alleen gegevens op die strikt noodzakelijk zijn voor de dienstverlening. Transactiegegevens worden bijvoorbeeld wel binnengehaald en verwerkt, maar we bewaren ze niet langer dan nodig. Standaard bewaren we detailtransacties lokaal op het toestel en alleen geaggregeerde of voor analyse nodige data op de server. Als er gevoelige info in omschrijvingen staat, kan deze worden gehashd of verborgen in opgeslagen vorm (tenzij nodig voor functies als categorie-herkenning).
* **Anonimisering:** Waar mogelijk anonimiseren we data. Bijvoorbeeld voor ontwikkelingsdoeleinden gebruiken we gesynthetiseerde of geanonimiseerde transactiedatasets. Logbestanden bevatten geen volledige persoons- of rekeninggegevens, maar bijv. alleen userIDs.
* **Controle door gebruiker:** De gebruiker heeft op ieder moment inzage in welke data Slim Minder van hem heeft. Via een **privacy-dashboard** in de accountinstellingen kan hij zien welke rekeningen gekoppeld zijn, welke categorie-info is opgeslagen, etc. Gebruikers kunnen tevens op ieder moment hun account (en bijbehorende data) laten verwijderen – dit wordt in GDPR-termijnen afgehandeld en we zorgen dat alle persoonlijke data dan daadwerkelijk uit onze systemen verdwijnt.
* **Toestemmingen:** Bij eerste installatie en bij het activeren van functies (zoals AI-coach die data gebruikt) vragen we duidelijk toestemming. De gebruiker kan bijv. kiezen of de AI zijn transactiehistorie mag inzien voor slimmere antwoorden, of dat de coach alleen generieke tips mag geven.
* **Dataverwerking locatie:** Alle data wordt verwerkt en opgeslagen op **EU-servers** (belangrijk ivm AVG). Mocht in de toekomst externe AI-diensten worden gebruikt die data buiten de EU verwerken, dan zullen we dit expliciet melden en waar mogelijk een EU-optie aanbieden.

In het kort: Slim Minder volgt het principe **“Privacy by design & by default”** – vanaf het ontwerpproces is privacy meegenomen zodat gebruikers zich veilig voelen hun financiële gegevens toe te vertrouwen.

**Schaalbaarheid en onderhoudbaarheid**

Hoewel we starten met een MVP, is vooruit gedacht aan groei en het gemak van onderhouden/updaten van de applicatie:

* **Microservices-architectuur:** In plaats van één logge applicatie, kiezen we voor een **modulaire opzet** (mits haalbaar in MVP-tijd). Bijvoorbeeld: een service voor transactieverwerking & categorisatie, een service voor notificaties, een service voor de AI-coach, etc. Deze microservices communiceren via duidelijke API’s (bijv. REST/JSON of GraphQL gateway) en kunnen onafhankelijk van elkaar worden geschaald en aangepast. Dit verhoogt de veerkracht: als de AI-service problemen heeft, blijft de rest van de app werken. Ook kunnen verschillende teamleden aan verschillende onderdelen werken zonder elkaar in de code te zitten.
* **Cloud-gebaseerde deployment:** We hosten Slim Minder in de **cloud** (AWS, Azure of GCP). Cloud biedt on-demand resources en **elasticity**: we kunnen vlot op- en afschalen bij meer gebruikers[thirdrocktechkno.com](https://www.thirdrocktechkno.com/blog/fintech-mvp/#:~:text=Cloud,hosting%2C%20data%20storage%2C%20and%20computation). We zetten containertechnologie in (Docker + Kubernetes mogelijk) zodat elke microservice geïsoleerd draait en eenvoudig gedistribueerd kan worden. **Load balancing** wordt ingericht om inkomende requests over meerdere instances te verdelen. We ontwerpen met **stateless** principes waar mogelijk, zodat horizontaal schalen simpel is.
* **CI/CD en DevOps:** Voor snelle iteratie en betrouwbaarheid richten we een **CI/CD-pipeline** in. Codewijzigingen doorlopen automatische build en test (unit/integratie) en kunnen daarna via geautomatiseerde deploy scripts naar een staging omgeving. Na goedkeuring kan dezelfde pipeline naar productie uitrollen. Door CI/CD houden we de releasecyclus kort en kunnen we frequent verbeteren zonder veel downtime. **Monitoring** (bijv. via Application Performance Monitoring tools) en logging zijn ingebouwd om issues in productie snel te spotten.
* **Onderhoudbaarheid van code:** We volgen best practices in coding standards (linting, code reviews) en documenteren de code intern. De keuze voor **React/React Native + Node.js** betekent we in één taal (JavaScript/TypeScript) veel development kunnen doen, wat kennisdeling bevordert. We structureren de codebase modulair en schaalbaar, gebruikmakend van moderne frameworks (bv. een Node framework als NestJS voor duidelijke architectuur). Dit alles om technische schuld te minimaliseren en toekomstige developers snel op weg te helpen.
* **Schaalbaarheid in ontwerp:** Bij het ontwerpen denken we ook aan toekomstige features: bijv. het dataschema is al voorbereid op meerdere bankaccounts per gebruiker, multi-user huishoudens, etc., ook al komt dat nu beperkt terug. We implementeren alleen MVP features nu, maar de **architectuur legt geen harde beperkingen op** die latere groei hinderen.

Met deze maatregelen kan Slim Minder bij succes groeien van enkele honderden testgebruikers naar potentieel duizenden of meer, zonder dat een complete herbouw nodig is. De **cloud-microservices-aanpak** biedt naadloze schaalbaarheid en hoge beschikbaarheid[thirdrocktechkno.com](https://www.thirdrocktechkno.com/blog/fintech-mvp/#:~:text=Cloud,hosting%2C%20data%20storage%2C%20and%20computation).

**Toegankelijkheid (Accessibility)**

We streven ernaar dat Slim Minder **voor iedereen toegankelijk** is, ongeacht beperkingen of leeftijd. Belangrijke aandachtspunten:

* **UI voor diverse doelgroepen:** De app is ontworpen mobile-first, maar met **responsive design** zodat hij op grotere schermen (web, tablets) ook goed werkt. We bieden in de instellingen opties voor **grote letters** en contrastrijke modus (voor slechtzienden of kleurenblinden). Alle belangrijke info is ook in tekst beschikbaar naast iconen/kleur (iemand die kleurenblind is, kan alsnog zien “budget 80% gebruikt” door tekstuele aanduiding).
* **Voice-over ondersteuning:** We testen de app grondig met schermlezers (bijv. VoiceOver op iOS, TalkBack op Android). Buttons en grafieken krijgen duidelijke **aria-labels** of beschrijvingen zodat een voice-over kan zeggen “Uitgaven deze maand grafiek, categorie Food: €200, 40% van totaal”. De chatfunctie van de AI-coach is ook te gebruiken via spraak tot tekst eventueel, zodat mensen met motorische beperkingen of analfabetisme er baat bij hebben.
* **Eenvoud en consistentie:** De navigatie is intuïtief, met herkenbare pictogrammen (home, grafiek, doel, chat) en consistente plaatsing. We vermijden ingewikkelde vaktermen en leggen functies uit in begrijpelijke taal. Waar nodig bieden we **hulpteksten** (bij een onbekend financieel begrip kan de gebruiker op een info-icoon klikken voor uitleg in simpel Nederlands).
* **WCAG standaarden:** We nemen de **WCAG 2.1 AA** richtlijnen als leidraad, wat betreft contrast, bedienbaarheid, begrijpelijkheid en robuustheid van de user interface. Hoewel misschien niet alle criteria voor MVP volledig gehaald worden, streven we ernaar zo dicht mogelijk bij volledige toegankelijkheid te zitten, zodat ook gebruikers met beperkingen Slim Minder kunnen bedienen.
* **Testen met echte gebruikers:** Onder de testaanpak valt ook toegankelijkheidstesten (zie verderop). Feedback van bijv. ouderen en laaggeletterden wordt verwerkt om de app verder te vereenvoudigen waar nodig.

Door deze focus zorgt Slim Minder dat ook kwetsbare groepen – die vaak juist baat hebben bij schuldpreventie – daadwerkelijk de app kúnnen gebruiken. Digitale inclusie is een belangrijk uitgangspunt.

**Taalondersteuning**

In de MVP fase is de app **volledig Nederlandstalig**. Alle teksten, meldingen en de AI-coach conversaties zijn in het Nederlands geschreven of getraind. Dit sluit aan bij de primaire doelgroep in Nederland. Echter, we houden er rekening mee dat in de toekomst uitbreiding naar andere talen of regio’s gewenst kan zijn:

* We bouwen de app met **internationalisatie (i18n)** in gedachten: teksten staan niet hardcoded verspreid in de code maar in centrale taalbestanden, zodat vertaling eenvoudig is. We structureren de content met keys, bijvoorbeeld t("dashboard.savings\_goal\_reached"), wat het later makkelijk maakt om een Engelse of Turkse vertaling toe te voegen.
* De AI-coach oplossing kiezen we zo dat meertaligheid mogelijk is. Bijvoorbeeld, OpenAI’s GPT-4 model ondersteunt al meerdere talen, dus een Engelse vraag zou het kunnen begrijpen (voor nu beperken we de UI tot NL). In de toekomst kunnen we de taal van de chat dynamisch instellen per gebruiker.
* Culturele aanpassingen: naast letterlijke taal, denken we ook aan locale instellingen. Bijvoorbeeld valutateken (€) en datum-/nummernotaties passen we aan per regio als we uitbreiden. In de MVP hard-coden we Euro en Nederlands formaat, maar de code kan eenvoudig aangepast worden om dit variabel te maken.

Kortom, de MVP focust op Nederlands, maar de technische basis is **uitbreidbaar naar meerdere talen** met relatief weinig moeite zodra de gelegenheid zich voordoet.

**Technische architectuur**

Hier beschrijven we de voorgestelde **technische architectuur** voor de Slim Minder MVP. We behandelen de keuze van de stack (frontend, backend, database, cloud), de integratie met PSD2 (bankkoppeling) en de AI-coach integratie. Het geheel is ontworpen voor een vlotte ontwikkeling van de MVP én toekomstige uitbreidbaarheid.

**Overzicht van de technologische stack**

* **Frontend (Mobiel & Web):** We gebruiken **React** als basis. Voor iOS en Android ontwikkelen we met **React Native**, zodat we één codebase in JavaScript/TypeScript hebben voor beide platforms. De webapp wordt gebouwd met **React** zelf (CRA of Next.js) waarbij we zoveel mogelijk componenten/herbruikbare logica delen met de React Native app (d.m.v. een monorepo of gedeelde util-packages). Styling gebeurt met een cross-platform compatibele methode (bijv. Styled Components of TailwindCSS voor web, en React Native StyleSheet voor mobiel). We zorgen voor een consistente UX over web en mobiel.
* **Backend:** We kiezen voor een **Node.js** backend, waarschijnlijk met een framework als **Express** of **NestJS** voor gestructureerde ontwikkeling. Node.js is gekozen vanwege de synergie met de React stack (zelfde taal) en goede ondersteuning voor real-time communicatie (handig voor push/notificaties). Alternatief had een JVM of .NET stack gekund, maar Node past goed bij een snelle MVP ontwikkeling. De backend wordt opgezet als een set van **microservices** (of in eerste instantie goed gescheiden modules binnen één service, met latere optie tot uitkoppelen). Voor MVP houden we aantal services beperkt: bijvoorbeeld één core API service (accounts, transacties, etc.), en aparte worker-processes voor intensievere taken (categorisatie jobs, AI-coach calls zodat frontend niet blokkeert). We implementeren een **REST API** (JSON gebaseerd) die door de frontends wordt aangesproken. Eventueel gevoelige acties (zoals triggering van bank refresh) kunnen via een **serverless function** of queue asynchroon verlopen om de responstijden naar de app kort te houden.
* **Database:** Voor dataopslag kiezen we een betrouwbare database. **PostgreSQL** is een goede kandidaat vanwege zijn sterkte in relationele data (gebruikers, accounts, transacties) en ondersteuning voor JSON (handig voor flexibele velden of storing van transactie details). Alternatief of aanvullend kan **MongoDB** dienen als een schema-flexibele store, maar gezien financiële data relationeel van aard is (transacties linken aan gebruikers, budgetten aan categorieën) is een SQL DB waarschijnlijk efficiënter. We kunnen gebruik maken van een ORM (zoals Prisma of Sequelize) om sneller te ontwikkelen en makkelijk te migreren. Voor de AI-coach conversaties of logs daarvan zouden we eventueel een aparte datastore kunnen hebben (bijv. een NoSQL if needed), maar in MVP kan dit ook in PostgreSQL.
* **Cloud Deployment:** We hosten alles in de cloud, bijvoorbeeld **AWS**. Daar draaien we de Node-services in containers (AWS ECS/Fargate of EKS voor Kubernetes, afhankelijk van complexity we aankunnen). De database kan draaien via een managed service (AWS RDS voor Postgres). Statische assets (de webapp build, images) hosten we via een CDN (AWS CloudFront/S3). We zetten een **CI/CD pipeline** op, bijvoorbeeld met GitHub Actions of GitLab CI, die test en deployt naar AWS. We configureren ook auto-scaling rules en monitoring (CloudWatch, etc). **Serverless functies**: eventueel gebruiken we AWS Lambda voor specifieke triggers (bijv. als PSD2 provider webhooks stuurt bij een nieuwe transactie, een Lambda ontvangt dat en zet het op onze queue).
* **API integraties:** Slim Minder zal communiceren met externe APIs:
  + **PSD2 API** van de gekozen open banking partner (Klarna/Invers/Salt Edge). Dit gebeurt via secure REST calls. Vaak levert de partner SDKs; we integreren dat op de backend zodat bankdata nooit direct naar de client gaat behalve via onze beveiligde server. De PSD2 integratie omvat zowel Account Information (voor transactie en saldo ophalen) als eventueel Payment Initiation (niet in MVP scope, maar wellicht toekomst). Voor MVP voornamelijk **AIS** (Account Information Service).
  + **Notificatie service:** voor het versturen van pushberichten op mobiel gebruiken we bijv. **Firebase Cloud Messaging (FCM)** of Apple APNs direct. De backend roept deze aan wanneer een nudge verstuurd moet worden. E-mail notificaties (voor rapporten of verificatie) via een service als SendGrid.
  + **AI/NLP API:** (zie aparte sectie hieronder).
* **Logging & Monitoring:** We integreren een oplossing voor logboek (bijv. Winston + ElasticSearch/Kibana stack of een cloud log service). Monitoring via metrieken (Prometheus/Grafana of een cloud alternative) zodat we inzicht hebben in performance, errors, etc. Alerts bij falen van kritieke componenten (zoals “PSD2 API niet bereikbaar” triggers een alert).

Deze stack is modern en veel gebruikt, waardoor we kunnen leunen op bestaande libraries en best practices. Dankzij React/React Native is de **gebruikerservaring consistent** en development efficiënt. Node.js en PostgreSQL in de cloud zorgen voor **robuuste server-side** mogelijkheden, met de flexibiliteit van JavaScript voor snelle aanpassingen.

**Integratie met PSD2-bankkoppeling**

Voor de bankkoppeling (PSD2) kiezen we een integratiepatroon dat **veilig, betrouwbaar en gebruikersvriendelijk** is. Een concreet voorstel is het gebruik van **Salt Edge** als aggregator (of een vergelijkbare partij als Klarna Kosma of Invers). Deze partijen bieden een **enkele API** waarmee we toegang krijgen tot bankgegevens bij vele banken, zonder voor elke bank apart iets te bouwen[saltedge.com](https://www.saltedge.com/products/data_enrichment#:~:text=The%20Data%20Enrichment%20platform%20can,on%20automatic%20account%20statement%20gathering).

**Hoe het werkt in de app:** Wanneer de gebruiker kiest om een bank te koppelen, wordt hij via onze app naar de **PSD2-aggregator flow** geleid (vaak in een in-app browser of redirect). Daar selecteert hij zijn bank en doorloopt de bank’s eigen authenticatie (bijv. inlog en een 2FA via de bankapp). Via OAuth2 ontvangt onze app (backend) vervolgens een **access token** die recht geeft om transacties op te halen. Deze token en refresh mechanismen slaan we encrypted op.

**Datastroom:** Met de verkregen toestemming kan onze backend via de aggregator API de rekeninginformatie ophalen: rekeningnummers, saldi en transactiehistorie (vaak laatste 90 dagen of meer). Dit doen we op initiële koppeling en vervolgens periodiek (bijv. 4x per dag of zo frequent als API toelaat) of via webhook als de aggregator dat ondersteunt. Transacties komen binnen als ruwe data (datum, bedrag, tegenrekening, omschrijving). Onze backend verwerkt ze: koppelen aan de juiste gebruiker, door de **categorisatie-engine** halen, en opslaan in de database.

**PSD2-verplichtingen:** Als Slim Minder zelf als AISP (Account Information Service Provider) optreedt, moeten we PSD2-vergunning hebben. Waarschijnlijk in MVP fase leunen we op de vergunning van de aggregator (die is geregistreerd bij De Nederlandsche Bank als PSD2 service). We zorgen wel dat alle juridische documentatie en user consent flows conform PSD2 zijn (duidelijke uitleg waarvoor data wordt gebruikt, etc.).

**Fallback:** Sommige banken ondersteunen PSD2 nog beperkt of de gebruiker vertrouwt het niet. Als fallback scenario kunnen gebruikers hun transactiegegevens importeren via een CSV of MT940 bestand. Dit is minder handig, maar voor MVP een vangnet.

**Keuze PSD2-partner aanbeveling:**

* *Salt Edge:* zeer breed (5000+ banken wereldwijd) en biedt naast data ook categorisatie, wat handig is[saltedge.com](https://www.saltedge.com/products/data_enrichment#:~:text=The%20categorisation%20API%20assigns%20a,by%20companies%20and%20their%20customers)[saltedge.com](https://www.saltedge.com/products/data_enrichment#:~:text=The%20Data%20Enrichment%20platform%20can,on%20automatic%20account%20statement%20gathering).
* *Klarna Kosma:* onderdeel van Klarna, focus op Europa, sterke API en recent actief op NL-markt.
* *Invers:* Benelux specialist, mogelijk betere lokale ondersteuning en expertise in Nederlandse banksystemen.

Voor MVP kiezen we de partij die de beste combinatie biedt van **dekkingsgraad** (alle grote NL banken), **ontwikkelaarsgemak** (goede docs, SDK) en **kosten**. In de implementatie abstraheren we de specifieke API zoveel mogelijk achter een service-laag, zodat we desgewenst kunnen wisselen van provider.

**AI-integratie (NLP-coach)**

De **AI-coach** functionaliteit vereist integratie met een **Natural Language Processing (NLP)** dienst voor het genereren van slimme antwoorden op gebruikersvragen. Gezien de MVP-tijdlijn is het aan te raden een bestaande AI-service te gebruiken in plaats van zelf een model te trainen. Mogelijke keuzes:

* **OpenAI GPT-4 API:** Zeer geavanceerd taalmodel, begrijpt en antwoordt in het Nederlands, en kan op basis van de juiste prompt financiële adviezen formuleren. We integreren via hun REST API: onze backend stuurt een prompt die bestaat uit de gebruikersvraag plus eventueel relevante context (bijv. “De gebruiker geeft 30% van inkomen uit aan huur, 20% aan boodschappen... [vraag:] Hoe kan hij besparen?”) en krijgt een antwoord terug. We moeten hier aandacht besteden aan het token-gebruik en kosten (OpenAI rekent per API call). We kunnen de prompt ook limiteren zodat er geen privacygevoelige exacte gegevens worden gedeeld, maar meer categorie-samenvattingen.
* **Alternatief: lokale/Nederlandse modellen:** Er zijn initiatieven voor Nederlandse taalmodellen (bijv. GPT-NL by Typetone[typetone.ai](https://www.typetone.ai/blog/gpt-nl-a-comprehensive-guide#:~:text=GPT,to%20understand%20and%20generate) of Bloembollen (Hypothetisch voorbeeld)). Deze kunnen on-premise of via een EU provider draaien, wat beter kan zijn voor privacy. Echter, vaak is hun kwaliteit (nog) lager dan GPT-4. Een middenweg is gebruik van een open-source model zoals GPT-J/Dolly 2.0 getuned op Nederlands, gehost in onze eigen cloud. Maar dit vraagt expertise en resources om operationeel te krijgen. Voor MVP geven we de voorkeur aan de **OpenAI route** om snel up-and-running te zijn.

**Architectuur AI-coach:** De AI-functionaliteit wordt geïsoleerd in een aparte module of service. De mobiele/web app communiceert niet rechtstreeks met OpenAI (dat zou het API key blootstellen), maar via onze backend: de frontend stuurt de vraag naar bijvoorbeeld /api/coach endpoint. De backend (coach-service) verrijkt de vraag met context (eventueel door even de DB te raadplegen voor de financiëlesamenvatting van de gebruiker) en doet de externe API call. Het antwoord wordt teruggegeven aan de app om weer te tonen in de chat UI.

We houden rekening met **latency**: zo’n API call kan ~1-3 seconden duren. We tonen daarom in de UI een “... is aan het typen”-indicator zodra de vraag verzonden is, om de wachttijd op te vangen.

**NLP voor Nederlandse context:** We moeten specifieke Nederlandse financiële termen en reallife context meegeven. Bijvoorbeeld termen als “zorgtoeslag”, “stufi” (studiefinanciering) zal de AI moeten begrijpen. Dit doen we door in de prompt een systeembericht mee te geven dat uitlegt: *“Jij bent een financiële coach voor Nederlandse gebruikers. Je bent bekend met Nederlandse financiële termen zoals toeslagen, IBAN, etc.”* Zo sturen we de AI.

**Content filtering:** We implementeren ook een check op de AI output. Als de AI een antwoord geeft dat potentieel ongepast of feitelijk onjuist is (bv. een risicovolle financiële tip), kunnen we ervoor kiezen het niet direct te tonen of het te voorzien van disclaimer. OpenAI API heeft wel ingebouwde content filtering, maar we blijven alert.

Samengevat, de AI-coach integratie voegt een innovatieve laag toe en maakt gebruik van krachtige bestaande NLP-modellen. De technische implementatie gebruikt **externe AI-API’s** veilig via onze backend, waarbij we zorgen voor Nederlandse context en naleving van privacy (geen onnodige persoonsgegevens in prompts).

**Roadmap MVP-oplevering**

Om een MVP vóór eind 2025 te realiseren, hanteren we een **gefaseerde aanpak** met duidelijke milestones en teamverantwoordelijkheden. Hieronder de roadmap met deliverables per fase:

**Fase 1: Concept & Architectuur (Q3 2024)**

* **Deliverables:** Voltooiing van dit technisch ontwerpdocument; definitieve keuze van tech-stack en PSD2-partner; opzet van basis architectuur (projectstructuur, repository, CI/CD pipeline). Daarnaast wireframes/prototypes van de belangrijkste schermen (dashboard, budget instellen, chatcoach) en uitwerking van de gebruikersstories in detail (als tickets).
* **Team & Rollen:** In deze fase zijn de **Product Owner/Business Analyst** en **Lead Architect** hoofdrolspelers. Zij werken samen om requirements scherp te krijgen. De **UI/UX Designer** levert de wireframes. Verder worden een **Scrum Master/Project Manager** en lead developers aangehaakt om planning en backlog op te zetten. Security/privacy expert bekijkt alvast het ontwerp voor compliance.

**Fase 2: Core Functionaliteit Ontwikkeling (Q4 2024 - Q1 2025)**

* **Deliverables:** Werkende implementatie van kernfunctionaliteiten: bankkoppeling (eenvoudige flow om 1 bank te koppelen en transacties op te halen), basis transactiecategorisatie, weergave transacties & categorie-saldo’s in de app, het kunnen instellen van budgetten per categorie en ontvangen van simpele waarschuwingen in de app (nog zonder push). Tevens eerste versie van het accountsysteem (registratie/inloggen, Free/Core onderscheid zonder alle beperkingen al actief).
* **Team & Rollen:** **Backend Developer(s)** richten de Node.js API in en integreren de PSD2-API. **Frontend/Mobile Developer(s)** bouwen de React (Native) schermen voor transacties en budgetinstelling. Een **Database Engineer** (of de backend dev) zet de PostgreSQL schema’s op. De **QA/tester** begint met basis testcases opstellen voor deze flows. Product Owner valideert tussentijds of de functionaliteit voldoet aan de gebruikerswensen.

**Fase 3: Insight & Nudging Features (Q2 2025)**

* **Deliverables:** Uitbreiding van functionaliteit: real-time **notificaties** via push (inrichten FCM/APNs), volledige **dashboard** met grafieken en maandrapport, implementatie van **spaardoelen** module (doelen aanmaken, voortgang bijhouden), en **gamification** elementen (badges behalen, eenvoudige challenges in de app configureren). In deze fase wordt ook de **AI-coach (chatbot)** geïntegreerd in een beta-vorm: de chatinterface werkt en een eerste versie AI geeft antwoorden op enkele voorgedefinieerde vraagtypes. Free/Core/Premium verschillen worden technisch geïmplementeerd (feature-toggling op basis van plan, bv. Free mag 1 doel, Premium onbeperkt, etc.).
* **Team & Rollen:** Naast de eerdere ontwikkelaars komt mogelijk een **AI/NLP Engineer** of externe consultant bij het team om de AI-coach integratie te doen (prompt design, API integratie). **DevOps Engineer** zet nu ook de productie-achtige omgeving op voor push notifications (VPN’s, etc. indien nodig) en zorgt voor schaalbaarheidstests. **UX Designer** finetunet de look-and-feel van grafieken en badges samen met frontenders. **QA** voert uitgebreidere tests uit, inclusief gebruikerstesten met een kleine groep (alpha/beta testers) om feedback te verzamelen.

**Fase 4: Polijsten, Veiligheid & Launch (Q3 2025)**

* **Deliverables:** In deze afsluitende fase richten we ons op **stabilisatie en kwaliteit**. Alle bekende bugs worden opgelost, de app krijgt performance tweaks waar nodig. **Beveiligingsfeatures** zoals 2FA, encryptie-audits, privacy-opties worden definitief gemaakt. De AI-coach wordt verder gevalideerd en waar mogelijk verbeterd (misschien met extra training data of regels). We voeren een **pilot test** met enkele eindgebruikers en eventueel een gemeente of partner, en verzamelen hun ervaringen. Ook worden **gebruikershandleidingen** of helpteksten afgerond. Uiteindelijk is de deliverable: een MVP versie v1.0 van Slim Minder opgeleverd aan eind 2025, gereed voor bredere uitrol.
* **Team & Rollen:** **Testers/QA** spelen hier een hoofdrol: ze doen regressietests, security testing (soms externe pentest laten doen), performance testing. **Developers** lossen laatste issues op en optimaliseren code. De **Security Officer/Expert** controleert naleving van GDPR en secure coding. **DevOps** zorgt dat de infrastructuur klaar is voor productie (monitoring ingesteld, backups geregeld). **Project Manager** coördineert de launch planning (bv. app store voorbereiden, web deployment).

**Post-MVP (indicatief)**

Hoewel buiten scope van deze MVP-planning, vermelden we ter context dat na MVP een fase van **evaluatie en iteratie** volgt (Q4 2025 en verder). Dan worden gebruikersfeedback en data geanalyseerd om de roadmap voor de volgende versie te bepalen. Mogelijk nieuwe features (multilingual, payment initiation om direct via app rekeningen te betalen, community-functies voor sociaal aspect, enz.) en opschalen naar meer gebruikers. De fundering die in MVP gelegd is, zal dit moeten kunnen dragen.

De bovenstaande roadmap zorgt dat we gefaseerd opleveren, met tussentijdse werkende software (we kunnen na fase 2 al een interne demo hebben, na fase 3 een gesloten beta). Elk teamlid weet per fase zijn verantwoordelijkheden, wat samenwerking en focus bevordert.

**Aanbevelingen voor testen en validatie**

Een degelijke test- en validatiestrategie is essentieel om de kwaliteit, veiligheid en gebruiksvriendelijkheid van Slim Minder te garanderen. We onderscheiden verschillende soorten testen:

* **Functionele tests (Unit/Integratie):** Vanaf het begin schrijven developers **unit tests** voor kritieke logica (bv. categorie-toekenning, berekenen budgetoverschrijding). **Integratietests** controleren of de keten werkt (van PSD2 data ophalen tot weergave in de UI). We maken gebruik van mocking voor externe API-calls (PSD2 sandbox, OpenAI sandbox) om scenario’s te simuleren. Deze tests draaien automatisch in de CI pipeline bij elke commit, zodat regressies snel opgemerkt worden[thirdrocktechkno.com](https://www.thirdrocktechkno.com/blog/fintech-mvp/#:~:text=B,Process).
* **Gebruikerstesten (UX/UI):** We plannen meerdere rondes van **gebruikersacceptatietesten** met mensen uit de doelgroep. Bijvoorbeeld, een groep jonge gebruikers test of zij de budget instel flow begrijpen; een groep ouderen test de leesbaarheid en navigatie. Hun feedback (denk aan onduidelijke termen, gemiste verwachtingen) verwerken we in UI-verbeteringen. Daarnaast doen we **usability testing** met hulpmiddelen: bijv. via een tool als Hotjar in de webapp om te zien waar men klikt of vastloopt (voor beta testers). Het doel is de app zo gebruiksvriendelijk en effectief mogelijk te maken voor release.
* **Security testing:** We onderwerpen de applicatie aan **security audits**. Dit omvat automatische scans (met OWASP ZAP of npm audit voor pakketten) en handmatige pentesting door een specialist. We testen o.a. of de API goed authenticatie afdwingt, of er geen ongeautoriseerde data access mogelijk is door ID-manipulatie, of we bestand zijn tegen SQL/NoSQL injection en XSS. Ook checken we het flows als bankkoppeling of 2FA op mogelijke misbruik. Eventuele bevindingen worden voor release verholpen. Bovendien stellen we een **responsible disclosure** beleid op voor het geval externe onderzoekers iets vinden.
* **AI-output validatie:** Omdat de AI-coach adviezen geeft, willen we zekerheid dat deze binnen correcte en ethische kaders blijven. We testen de AI met een reeks **voorbeeldvragen** (normaal, randgevallen en ook ongepaste vragen) en beoordelen de antwoorden. We letten op feitelijke juistheid (geen verkeerde financële info), toon (begrijpelijk, niet veroordelend) en of er biases zijn. Indien nodig passen we de prompt aan of stellen we filters in (bijv. als iemand vraagt om illegale financiële adviezen zal de AI niet antwoorden). Na lancering monitoren we de AI-gesprekken (geanonimiseerd) om te zien hoe het presteert en vangen we potentiële problemen op. Users krijgen bovendien de optie om feedback te geven op coach-antwoorden (“Was dit nuttig? Ja/Nee”), zodat we kunnen bijsturen.
* **Prestatie- en schaalbaarheidstesten:** Voor go-live doen we een **load test** van kritieke onderdelen. Bijvoorbeeld, simuleren van 1000 gelijktijdige gebruikers die notificaties ontvangen of de AI vragen stellen, om te kijken of de servers het aankunnen en waar bottlenecks zitten. We gebruiken tools als JMeter or k6 voor API load, en wellicht Firebase testlab for mobile performance. Eventuele performance issues (memory leaks, trage queries) lossen we op. We streven naar een lichte app (< X MB download), snelle laadtijden en soepel gebruik, ook op wat oudere devices (we testen bv. op een 3 jaar oude Android telefoon om te zien of animaties nog oké zijn).
* **Toegankelijkheidstests:** Zoals genoemd laten we mensen met bijv. visuele beperkingen de app testen (in samenwerking met organisaties of via netwerk). We gebruiken ook softwaretools (bijv. lighthouse, axe) om de webapp te scannen op toegankelijkheidsproblemen (contrast ratios, alt-teksten, keyboard navigatie). Issues zoals te laag contrast of ontbrekende labels worden gefixt voor release. Dit zorgt ervoor dat we de WCAG-richtlijnen praktisch getest hebben.
* **Beta test & feedback loop:** Tot slot, we raden aan om de MVP eerst als **beta** uit te rollen onder een beperkte groep (bijv. via TestFlight voor iOS, beta channel for Android, en een afgeschermde web login). In deze fase verzamelen we echte gebruiksdata en eventuele crash logs via tools (Sentry, Crashlytics). We houden een supportkanaal open (bijv. via Discord of e-mail) voor betatesters om issues te melden. Deze laatste feedbackronde verwerken we in een snelle update voordat we de brede lancering doen.

Concluderend, door een mix van automatische tests, handmatige audits en echte gebruikersfeedback[thirdrocktechkno.com](https://www.thirdrocktechkno.com/blog/fintech-mvp/#:~:text=B,Process) zorgen we dat Slim Minder MVP niet alleen functioneel rijk is, maar ook **betrouwbaar, veilig en gebruiksvriendelijk**. Elk onderdeel – van de PSD2-koppeling tot de AI-coach – wordt gevalideerd, zodat het technische team en stakeholders met vertrouwen de app kunnen lanceren als een solide eerste versie van Slim Minder.