

DELTA – Střední škola informatiky a ekonomie, s.r.o.

Ke Kamenci 151, 530 03 Pardubice

Bible Memorizer

(Gamifikovaná platforma pro retenci textů)

Autor: Matěj Komárek

Obor: Informační technologie

Třída: 4. B

Školní rok: 2025/2026

Zadání maturitního projektu z informatických předmětů

Jméno a příjmení: *Matěj Komárek*
Školní rok: *2025/2026*
Třída: *4.B*
Obor: *Informační technologie 18-20-M/01*
Téma práce: *Mobilní aplikace pro zapamatování biblických veršů*
Vedoucí práce: *Ing. Monika Borkovcová, Ph.D.*

Způsob zpracování, cíle práce, pokyny k obsahu a rozsahu práce:

Cílem projektu je vytvořit mobilní aplikaci pro operační systém Android, která uživatelům pomůže zapamatovat si biblické verše efektivním a zároveň zábavným způsobem. Důraz je kladen na motivaci a dlouhodobou angažovanost uživatele.

Aplikace bude obsahovat tyto funkce:

- Zobrazení textu Bible - možnost výběru překladu a označení veršů k učení.
- Učení pomocí miniher – například doplňování chybějících slov, skládání slov ve správném pořadí, apod.
- Opakovací algoritmus založený na principu spaced repetition – automatické plánování opakování veršů pro efektivní zapamatování.
- Správa uživatelského účtu – registrace, přihlášení a uchování pokroku.
- Sociální funkce – možnost přidat přátele, sledovat jejich statistiky a zobrazit porovnání v žebříčku
- Statistiky a vizualizace pokroku – přehled naučených veršů, doby učení a denní řady
- Notifikace – připomenutí pro opakování veršů nebo pokračování v učení.
- Offline režim – základní funkce aplikace budou dostupné i bez připojení k internetu.
- Bezpečnostní opatření a zabezpečení dat, včetně autentifikace, autorizace a šifrování datových přenosů.
- Možnost nastavení uživatelských preferencí a personalizace prostředí aplikace.
- Optimalizace pro různé typy zařízení, aby byla aplikace použitelná a přístupná pro co nejširší spektrum uživatelů.

Práce může být vypracovaná anglickém jazyce.

Stručný časový harmonogram (s daty a konkretizovanými úkoly):

Září: Analýza existujících aplikací s podobným zaměřením, definování požadavků cílové skupiny, návrh architektury a rozhraní aplikace.

Říjen - Leden: Implementace a vývoj mobilní aplikace a backendu s důrazem na plnění definovaných požadavků. Testování a ladění funkcionality aplikace.

Leden: Zahájení zpracování dokumentace aplikace a teoretické části práce.

Únor - Březen: Dokončení zpracování dokumentace a teoretické části práce. Dolad'ování a oprava chyb v aplikaci na základě testování. Příprava na prezentaci a obhajobu práce.

Dne 25. 6. 2025



.....
podpis ředitele

převzal:

Dne 7.9. 2025

Komárek
.....
datum a podpis žáka

Prohlašuji, že jsem maturitní projekt vypracoval(a) samostatně, výhradně s použitím uvedené literatury.

V Pardubicích 30. 3. 2026

(vlastnoruční podpis)

Upřímně děkuji Ing. Monice Borkovcové, Ph.D. jak za odborné vedení a cenné rady při zpracování tohoto projektu, tak za trpělivost a podporu během celé práce.

Resumé

Práce dokumentuje návrh a vývoj mobilní aplikace Bible Memorizer zaměřené na memorizaci biblických textů. Realizace projektu zahrnuje mobilní aplikaci s podporou offline režimu a backendový server pro synchronizaci a sociální funkce. Mobilní aplikace dále podporuje gamifikaci a učení pomocí intervalového opakování.

Klíčová slova

Programování, gamifikace, intervalové opakování, učení, Flutter, NestJS, Bible, memorizace

Abstract

This thesis documents the design and development of the Bible Memorizer mobile application, which focuses on the memorization of biblical texts. The implementation of the project includes a mobile application with offline mode support and a backend server for synchronization and social features. Furthermore, the mobile application supports gamification and learning through spaced repetition.

Keywords

Programming, gamification, spaced repetition, learning, Flutter, NestJS, Bible, memorization

Obsah

| | |
|--------------------------------------|-----------|
| Úvod | 9 |
| 1 Terminologie | 10 |
| 1.1 Verš | 10 |
| 1.2 Intervalové opakování | 10 |
| 2 Analýza existujících řešení | 11 |
| 3 Analýza požadavků | 12 |
| 3.1 Funkční požadavky | 12 |
| 3.2 Nefunkční požadavky | 13 |
| 4 Technologie | 14 |
| 4.1 Flutter | 14 |
| 4.2 NestJS | 14 |
| 4.3 PostgreSQL | 14 |
| 4.4 SQLite | 14 |
| 4.5 Cloudflare Tunnel | 15 |
| 5 Architektura | 16 |
| 5.1 Mobilní aplikace | 16 |
| 5.1.1 Ukládání dat | 16 |
| 5.1.2 Algoritmy | 16 |
| 5.2 Server | 17 |
| 5.2.1 Cloudflare Tunnel | 17 |
| 5.2.2 Synchronizace | 17 |
| 6 Aplikace | 18 |
| 6.1 Datový model | 18 |
| 6.1.1 Entity datového modelu | 18 |
| 6.1.2 Zajištění integrity a relace | 19 |
| 6.2 Bezpečnost | 20 |
| 6.2.1 Autentizace a ochrana identity | 20 |
| 6.2.2 Šifrování a síťová bezpečnost | 20 |

| | | |
|----------|---|-----------|
| 6.2.3 | Autorizace a ochrana dat | 20 |
| 6.3 | Dokumentace API rozhraní | 21 |
| 6.3.1 | Synchronizační endpointy (SyncController) | 21 |
| 6.3.2 | Uživatelské endpointy (UserController) | 22 |
| 6.3.3 | Systémové endpointy (AppController) | 23 |
| 6.4 | Uživatelské rozhraní | 23 |
| 6.4.1 | Přístupnost a personalizace | 24 |
| 6.5 | Gamifikace a motivační prvky | 25 |
| 7 | Shrnutí výsledků | 28 |
| | Závěr | 29 |
| | Literatura | 30 |
| | Seznam příloh | 33 |
| | Přílohy | 34 |

Úvod

Ve světě žije přes 2.3 miliardy křesťanů [2], což znamená, že více než čtvrtina lidí jsou křesťané. Častým cílem křesťanů je studium Bible a zapamatování jejích částí. Dlouhodobé zapamatování textu ale není jednoduchý úkol. Existují aplikace, které memorizování částí Bible zjednodušují, ale ty jsou často zastaralé, těžké na používání a neaplikují metody učení, které uživateli pomáhají, urychlují a zpříjemňují celý proces.

Za tímto účelem vznikla aplikace Bible Memorizer. Tato aplikace se snaží jejím uživatelům zjednodušit cíl zapamatování specifických biblických testů. K tomu používá dva hlavní principy. První z nich je gamifikace. Aplikace používá rozmanité minihry, které uživatele přirozeně nutí zastavit se nad daným textem a přemýšlet nad ním novým způsobem.

Druhý princip je intervalové opakování. Intervalové opakování je způsob memorování informace na základě postupně rostoucích intervalů. Díky tomu se uživatelem naučený text uživateli zobrazuje pouze občasně pro kontrolu jeho retence. Uživatel pak nemusí trávit tolik času opakováním veršů, které jsou pro něj jednoduché.

Jak lze z celé koncepce projektu jednoduše odvodit, cílovou skupinou této aplikace jsou výhradně křesťané, kteří si chtějí memorizovat biblické verše.

Aplikace není vázána na specifickou denominaci či církev.

Aplikace zároveň podporuje lokalizaci, díky čemuž umožňuje používání křesťanům z jiných zemí, než jen Česká republika. (Aplikace momentálně podporuje český a anglický jazyk)

1 Terminologie

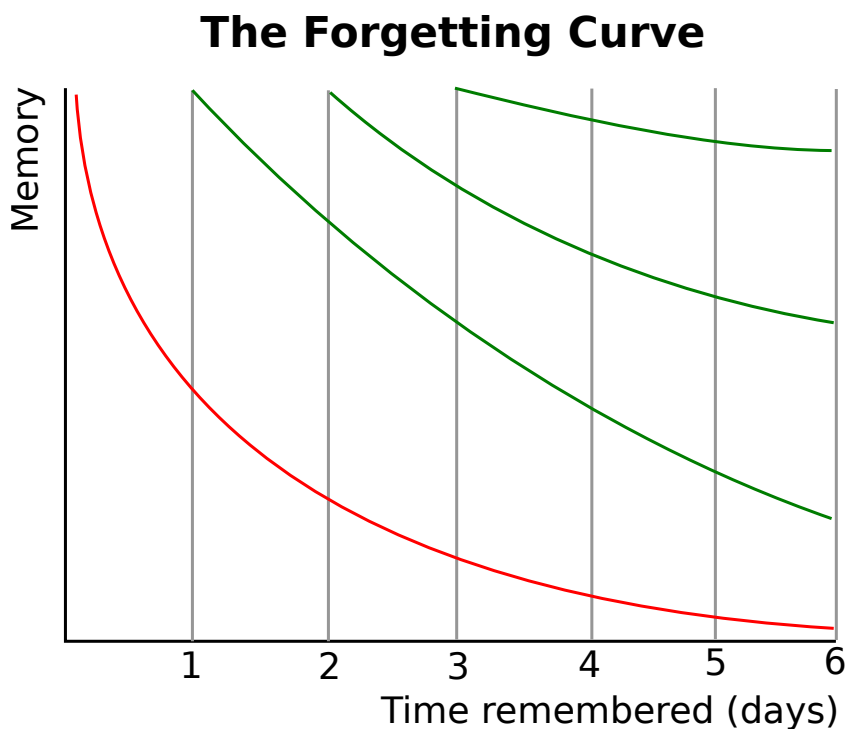
1.1 Verš

Verš je část textu Bible. Jeho délka je většinou jedna až dvě věty. Jednotlivé verše tvoří kapitoly a jednotlivé kapitoly tvoří biblické knihy, které pak tvoří Bibli.

Každý verš má svojí unikátní, neměnnou referenci většinou udávanou ve formě "název-knihy číslo-kapitoly:číslo-verše". (např. "Jan 3:16")

1.2 Intervalové opakování

Intervalové opakování [8] je způsob učení, který se zakládá na postupně rostoucích intervalech. Vysvětleme si to na příkladu. Student se chce naučit nazpaměť jeden verš. První den se ho naučí. Druhý den si ho zopakuje. Třetí den vynechá, ale čtvrtý den se ujistí, že si ho stále pamatuje. Tento interval postupně roste, takže student si danou informaci neopakuje každý den. Díky tomu může získaný čas využít na jiné učivo a pojmout tak více informací. Ebbinghausova křivka zapomínání [3] ukazuje, jak rychle člověk zapomíná informaci. Jak můžeme vidět na obrázku 1, tak daná křivka (v červené barvě) se zplošťuje s rostoucím počtem opakování.



Obrázek 1: Křivka zapomínání [1]

2 Analýza existujících řešení

V rámci rešerše trhu a mapování konkurenčního prostředí byly podrobně analyzovány čtyři existující mobilní aplikace: VerseLocker [4], Bible Memory [5], Remember Me [6] a BibleMe [7]. Ačkoliv je primární cíl těchto nástrojů totožný – pomoci uživateli s memorizací textu – analýza odhalila řadu kritických nedostatků, které otevírají prostor pro vytvoření nového, modernějšího řešení.

Zásadním zjištěním je, že většina analyzovaných aplikací nevyužívá pokročilé metody učení, jako je například intervalové opakování (spaced repetition). Učení se tak stává méně efektivním. Dalším častým problémem je zastaralé a neintuitivní uživatelské prostředí (UI), které novým uživatelům ztěžuje orientaci a snižuje celkový zážitek z používání (UX).

Z hlediska práce s biblickým textem je častým nedostatkem absence integrované čtečky. Aplikace se často spoléhají na textové pole, do kterého je nutné manuálně zadat přesnou referenci (např. „Jan 3:16“). Tento přístup je náchylný k chybám, uživatelsky nepřívětivý a v jedné z testovaných aplikací byl tento způsob vyhledávání dokonce zcela nefunkční.

Analýza se zaměřila i na motivační a sociální prvky. Pouze jedno z testovaných řešení obsahovalo systém uživatelských žebříčků. Tento systém byl však navržen jako globální a hodnotil uživatele na základě celkového počtu „zapamatovaných“ veršů. Vzhledem k tomu, že stav zapamatování si určuje sám uživatel bez jakéhokoliv objektivního testování, je tento systém snadno zneužitelný a ve výsledku působí spíše demotivačně.

V neposlední řadě představuje velkou bariéru pro tuzemské uživatele absence lokalizace. Žádná z testovaných aplikací nenabízela uživatelské prostředí v českém jazyce a pouze jedna umožňovala stažení českého překladu Bible. Možnosti synchronizace dat byly navíc u většiny aplikací omezené pouze na e-mail, případně na sociální síť Facebook.

3 Analýza požadavků

Na základě zjištění z analýzy existujících řešení a definování cílové skupiny byly stanoveny konkrétní požadavky na vyvíjenou aplikaci. Tyto požadavky se dělí na funkční, které popisují chování systému, a nefunkční, které definují jeho vlastnosti a kvalitu.

3.1 Funkční požadavky

Systém musí uživatelům poskytovat následující funkcionalitu:

- **Správa uživatelského účtu:** Systém musí umožňovat registraci a přihlášení pomocí účtu Google, odhlášení a kompletní smazání uživatelského účtu i s přidruženými daty.
- **Práce s biblickým textem:** Aplikace musí zobrazovat text Bible s možností plynulého čtení a změny překladu.
- **Správa veršů k memorizaci:** Uživatel musí mít možnost uložit si verš k zapamatování, a to přímým výběrem z textu Bible.
- **Modul učení:** Aplikace musí nabízet procvičování konkrétního vybraného verše i automaticky generovanou sadu doporučených veršů na základě algoritmu intervalového opakování.
- **Zobrazení statistik:** Systém musí uživateli vizualizovat jeho osobní statistiky a postup v učení.
- **Sociální modul:** Uživatelé musí mít možnost přidávat a odebírat přátele (pomocí unikátního kódu), prohlížet si jejich statistiky a porovnávat své výsledky v žebříčku.
- **Personalizace a nastavení:** Aplikace musí umožňovat změnu jazyka rozhraní, přepínání mezi světlým a tmavým motivem (light/dark mode) a úpravu velikosti textu.

Komplexní přehled všech interakcí uživatele se systémem je zachycen v UML diagramu případů užití, který je součástí Přílohy 1.

3.2 Nefunkční požadavky

Aby byla zajištěna vysoká kvalita a využitelnost softwaru, musí systém splňovat následující kritéria:

- **Výkon a kompatibilita:** Aplikace musí být optimalizována pro plynulý běh i na starších zařízeních, musí podporovat různá rozlišení a poměry stran displeje. Minimální podporovaná verze operačního systému je Android 10.
- **Spolehlivost a dostupnost (Offline režim):** Základní funkce aplikace, jako je prohlížení uložených veršů, hraní miniher a opakování, musí být plně dostupné i bez aktivního připojení k internetu. Uživatelská data se musí automaticky synchronizovat se serverem po obnovení připojení.
- **Bezpečnost:** Veškeré datové přenosy mezi klientem a serverem musí být šifrovány. Systém musí implementovat spolehlivou autentizaci uživatelů a striktní autorizaci pro zajištění bezpečného oddělení dat mezi jednotlivými účty.
- **Použitelnost a přístupnost:** Uživatelské prostředí musí být intuitivní. Důraz je kladen na motivaci a dlouhodobou angažovanost (gamifikace). Aplikace musí být přístupná širokému spektru uživatelů, včetně osob se zhoršeným zrakem (podpora změny velikosti písma).

4 Technologie

4.1 Flutter

Pro vývoj klientské aplikace byl zvolen framework Flutter. [9] Vzhledem k implementaci interaktivních miniher a nutnosti plynulých animací při gamifikaci bylo klíčové zajistit konzistentní vykreslování napříč různými verzemi systému Android. Vlastní renderovací engine Flutteru tento problém řeší tím, že se nespolehá na nativní komponenty OEM výrobců, čímž eliminuje vizuální fragmentaci a snižuje náročnost testování UI. Zároveň nativní podpora asynchronních operací (Dart) usnadnila implementaci náročného lokálního vyhodnocování algoritmu SM-2 bez blokování uživatelského rozhraní.

4.2 NestJS

Serverová logika je postavena na frameworku NestJS. [10] Pro projekt typu Bible Memorizer, který vyžaduje složitou obousměrnou synchronizaci offline dat, bylo nutné zajistit striktní oddělení business logiky. Architektura NestJS založená na závislosti (Dependency Injection) umožnila bezpečně izolovat modul pro autentizaci uživatelů od samotného synchronizačního jádra (SyncController). Toto rozdělení usnadnilo implementaci mechanismu 'zdroje pravdy' (source of truth) pro řešení datových konfliktů při slučování časových rázitek, aniž by došlo k narušení ostatních doménových procesů.

4.3 PostgreSQL

Jako primární databázový systém pro běh na serveru je nasazen PostgreSQL. [11] Slouží k trvalému ukládání relačně strukturovaných uživatelských dat. Podpora komplexních relací a striktní dodržování pravidel transakčního zpracování zajišťují absolutní datovou integritu a konzistenci při souběžných aktualizacích a synchronizačních požadavcích od vícera klientů.

4.4 SQLite

Pro správu lokálních dat přímo v paměti mobilního zařízení je využita databáze SQLite. [12] Její implementace je nezbytnou podmínkou pro fungování aplikace v offline režimu. Entitní data o uložených verších, harmonogramu opakování a pokroku v učení jsou ukládána do tohoto lokálního úložiště a následně, po detekci obnoveného síťového připojení,

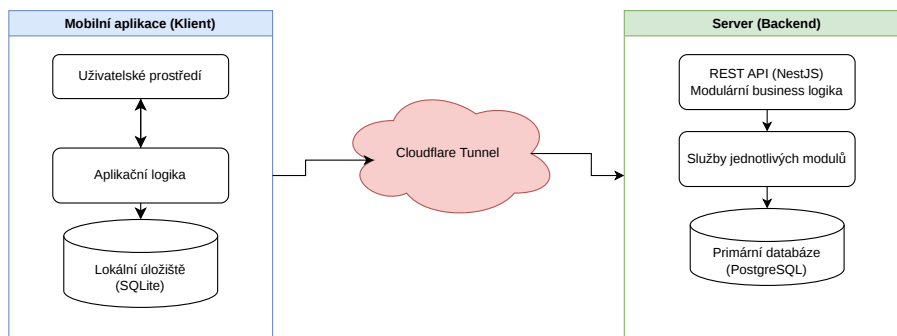
asynchronně synchronizována s hlavní databází na serveru. Uživatel tak neztrácí kontinuitu učení ani při výpadech sítě.

4.5 Cloudflare Tunnel

Infrastrukturní spojení mezi lokálním serverem a sítí internet zajišťuje démon Cloudflare Tunnel. [13] Tato technologie vytváří šifrované odchozí spojení z aplikačního kontejneru přímo do sítě Cloudflare. Serverová aplikace (NestJS) díky tomu může bezpečně přijímat vnější požadavky bez nutnosti otevírání veřejných portů na serverovém firewallu nebo nutnosti přidělení statické veřejné IP adresy, čímž se minimalizuje povrch pro potenciální síťové útoky.

5 Architektura

Architektura systému je navržena jako klient-server, kde klientskou část tvoří mobilní aplikace a logiku zpracování dat centralizuje serverová část. Tyto dvě komponenty spolu komunikují převážně asynchronně, aby byla zachována vysoká dostupnost a možnost offline fungování. Celkové schéma propojení klientské aplikace, zabezpečeného tunelu a serverového backendu je zobrazeno na obrázku 2.



Obrázek 2: Architektura projektu

5.1 Mobilní aplikace

Klientská část je zodpovědná za prezentaci uživatelského rozhraní, lokální vyhodnocování algoritmů a dočasnou perzistenci dat.

5.1.1 Ukládání dat

Lokální ukládání dat v mobilní aplikaci je primárně zajištěno pomocí relační databáze SQLite. Toto řešení je klíčové pro zajištění plnohodnotného běhu aplikace v offline režimu, kdy si uživatel může prohlížet a procvičovat uložené verše bez závislosti na momentálním internetovém připojení. Pro účely následné synchronizace je v lokálním databázovém schématu zavedena vlastnost (*needsSync*), která označuje ty databázové záznamy, jež byly na zařízení nově vytvořeny nebo modifikovány a čekají na odeslání na backend.

5.1.2 Algoritmy

Jádrem logiky aplikace pro plánování učení je implementace principu intervalového opakování (Spaced Repetition). Konkrétně je využit algoritmus SM-2 (SuperMemo-2) [14], který na základě předchozího hodnocení obtížnosti daného verše a zaznamenané úspěš-

nosti při aktuálním procvičování dynamicky vypočítává datum dalšího opakování. Tímto přístupem dochází k optimalizaci procesu učení – problematické verše jsou uživateli k procvičení předkládány v kratších intervalech, zatímco u dobře osvojených textů se interval do dalšího zobrazení exponenciálně prodlužuje.

5.2 Server

Serverová část aplikačního řešení poskytuje REST API a spravuje primární databázové úložiště. Backend je postaven na frameworku NestJS. Z architektonického hlediska je aplikační kód rozdělen do logických modulů, což umožňuje systematické zapouzdření jednotlivých doménových funkcionalit. Toto modulární členění zajišťuje izolaci business logiky a bezpečné zpracování požadavků.

Logická struktura tříd backendové aplikace, jejich metody a vzájemné vztahy jsou detailně zobrazeny v UML diagramu tříd v Příloze 2.

5.2.1 Cloudflare Tunnel

Pro bezpečné vystavení lokálního serveru do prostředí sítě internet je využit nástroj Cloudflare Tunnel. Tato technologie vytváří zabezpečené odchozí spojení přímo do sítě Cloudflare, čímž eliminuje nutnost otevírání veřejných portů na serverovém firewallu či přiřazování statických veřejných IP adres. Zásadním přínosem tohoto řešení je nativní zajištění šifrované komunikace, kdy je veškerý datový provoz mezi klientskou aplikací a serverem automaticky směrován přes protokol HTTPS, což zaručuje důvěrnost a integritu přenášených dat.

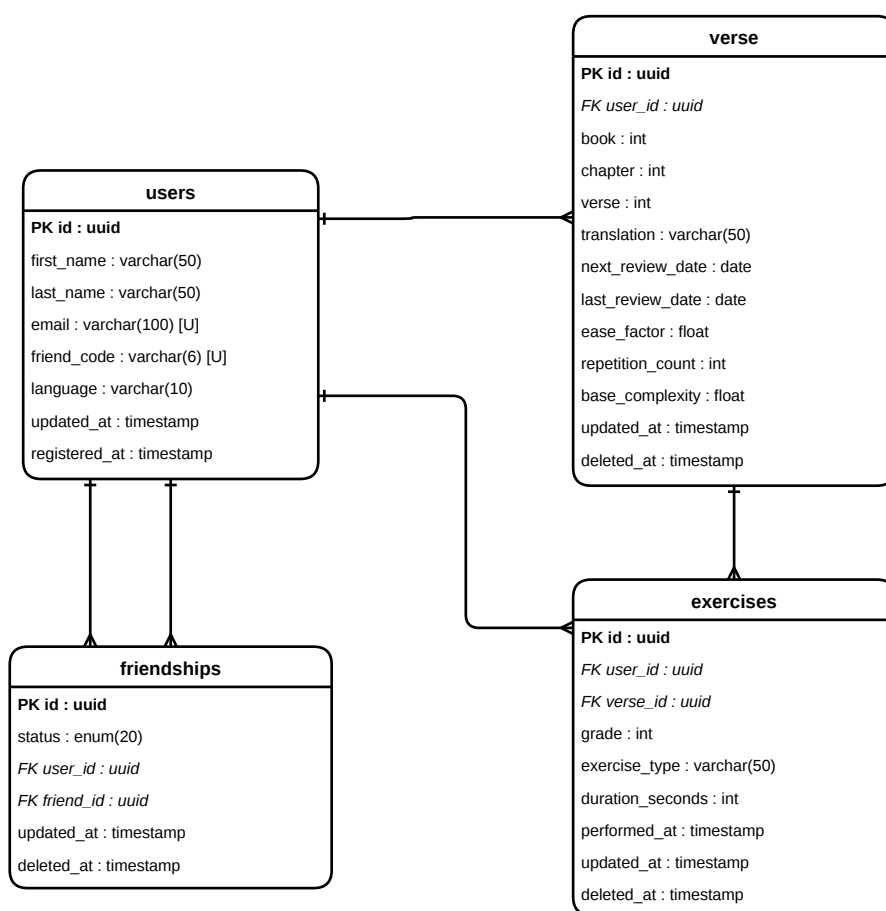
5.2.2 Synchronizace

Synchronizační mechanismus zajišťuje konzistenci dat mezi lokální databází na mobilním zařízení a hlavní serverovou databází. Proces synchronizace je iniciován ze strany klienta. Aplikace vyfiltruje záznamy s vlastností *needsSync* a odešle je formou dávky na server. Pro řešení případných datových konfliktů – například pokud uživatel procvičoval verše offline na dvou různých zařízeních – je implementován mechanismus „zdroje pravdy“ (source of truth) založený na časovém razítku poslední úpravy záznamu. Server při slučování dat porovná časová razítka (*timestamp* nebo datum procvičení) a do hlavní databáze zapíše tu variantu záznamu, která nese nejnovější časový údaj.

6 Aplikace

6.1 Datový model

Pro zajištění perzistence dat a spolehlivého chodu aplikace byla navržena robustní databázová architektura. Architektura využívá kombinaci lokální databáze SQLite na mobilním zařízení a centrální relační databáze PostgreSQL na serveru. Následující datový model reflektuje strukturu hlavní serverové databáze. Struktura serverové databáze, včetně definic primárních a cizích klíčů, je vizualizována v ER diagramu na obrázku 3.



Obrázek 3: Datový model

6.1.1 Entity datového modelu

Databázové schéma je rozděleno do čtyř klíčových entit, které reprezentují základní domény aplikace:

- **Uživatel (users):** Centrální entita celého systému. Uchovává identifikační údaje uživatele (jméno, příjmení, e-mail) a jeho osobní preference v aplikaci, jako je zvolený jazyk rozhraní. E-mailová adresa slouží jako unikátní identifikátor a nesmí se v systému duplikovat.
- **Uložený verš (saved _verses):** Entita reprezentující konkrétní biblický text, který si uživatel zvolil k memorizaci. Obsahuje lokační údaje textu (kniha, kapitola, verš) a zvolený překlad. Pro účely intervalového opakování jsou zde uloženy klíčové dynamické atributy: datum posledního procvičení, datum dalšího naplánovaného procvičení a další údaje využívané pro algoritmus intervalového opakování. Každý verš je striktně vázán na právě jednoho uživatele.
- **Procvičení (exercises):** Transakční entita sloužící k logování historie učení. Vzniká při každém dokončení minihry. Zaznamenává časový údaj (timestamp) provedení, celkovou dobu trvání cvičení v sekundách, typ zvolené minihry a hodnotu reprezentující úspěšnost. Záznam je relačně propojen s konkrétním uživatelem a konkrétním uloženým veršem.
- **Přátelství (friendships):** Vazební entita realizující sociální funkce aplikace. Definuje vztah mezi dvěma různými uživateli. Obsahuje stav žádosti. Součástí je také časové razítko vytvoření vazby.

6.1.2 Zajištění integrity a relace

Aby byla zaručena absolutní konzistence dat a ochrana proti anomáliím, využívá databázový model soustavu integritních omezení:

- **Referenční integrita a kaskádové mazání:** Všechny cizí klíče (vazby na uživatele či verše) jsou definovány s pravidlem ON DELETE CASCADE. Pokud je odstraněn uživatelský účet, databáze automaticky a bezpečně odstraní i všechny jeho uložené verše, historii procvičování a vazby přátelství. Tím je zamezeno vzniku tzv. sirotčích záznamů.
- **Unikátní omezení:** Tabulka přátelství obsahuje unikátní omezení pro dvojici uživatelů, aby nedocházelo k duplicitním žádostem. Rovněž je pomocí omezující podmínky zakázáno, aby uživatel odeslal žádost o přátelství sám sobě.

- **Ověření domén:** V databázi jsou definována přísná pravidla pro vkládané hodnoty. Číslo kapitoly, číslo verše a délka procvičení musí být vždy kladná čísla.
- **Indexace:** Pro zajištění vysokého výkonu při synchronizaci a vyhledávání jsou nad všemi cizími klíči (např. identifikátor uživatele v tabulce historie) vytvořeny databázové indexy.

6.2 Bezpečnost

Vzhledem k povaze ukládaných uživatelských dat (osobní e-maily, statistiky učení, seznamy přátel) byl kladen vysoký důraz na implementaci adekvátních bezpečnostních opatření, a to jak na straně klientské aplikace, tak na straně serverové infrastruktury.

6.2.1 Autentizace a ochrana identity

Proces ověřování identity uživatelů (autentizace) je řešen výhradně pomocí Google účtu. Tento přístup přenáší odpovědnost za bezpečné ukládání hesel na poskytovatele třetí strany a uživateli nabízí pohodlnější a bezpečnější formu přístupu bez nutnosti vytvářet nové přihlašovací údaje.

Pro sociální funkce (přidávání přátel) byl zaveden systém unikátních šestimístných alfa-numerických identifikátorů (Friend Code). Díky tomu je při vyhledávání jiných uživatelů zcela skryta jejich skutečná e-mailová adresa.

6.2.2 Šifrování a síťová bezpečnost

Veškerá komunikace mezi mobilním zařízením a serverem podléhá striktnímu šifrování v průchodu (encryption in transit). Toho je docíleno nasazením technologie Cloudflare Tunnel. Serverová komponenta díky tomuto řešení nemusí mít vystavené veřejné porty, čímž se drasticky snižuje riziko přímého útoku na infrastrukturu. Veškerý datový provoz je tak nuceně směrován přes zabezpečený protokol HTTPS, který zajišťuje důvěrnost a zabraňuje zachycení komunikace.

6.2.3 Autorizace a ochrana dat

Serverová architektura postavená na frameworku NestJS využívá propracovaný systém kontrolerů a guardů (ochranných vrstev). Ty zajišťují, že každý požadavek směřující na

aplikační programové rozhraní (API) je podroben kontrole přístupových práv. Uživatel může prostřednictvím synchronizačních endpointů číst, upravovat či odstraňovat výhradně ty databázové záznamy, které jsou jednoznačně přiřazeny k jeho vlastnímu identifikačnímu číslu (`userId`). Jakýkoliv pokus o manipulaci s daty jiných uživatelů je serverem automaticky zamítnut.

6.3 Dokumentace API rozhraní

Backendová část aplikace postavená na frameworku NestJS poskytuje REST API, které je navrženo pro efektivní synchronizaci dat a správu uživatelských profilů. Veškerá komunikace (s výjimkou systémových endpointů) vyžaduje autentizaci pomocí JWT tokenu.

6.3.1 Synchronizační endpointy (`SyncController`)

Tyto endpointy tvoří jádro aplikace a umožňují dávkové (`batch`) zpracování dat pro zajištění plynulého offline režimu.

- **GET `/sync/pull`**
 - **Popis:** Slouží k získání dat ze serveru směrem do klientské aplikace od času poslední úspěšné synchronizace.
 - **Vstupní parametry (Query):** `lastSync` (String, ISO datum – čas poslední úspěšné synchronizace).
 - **Návratová hodnota:** JSON objekt obsahující `timestamp` (aktuální čas serveru) a objekt `changes` se seznamy nových či upravených entit (`verses`, `exercises`, `friendships`, `user`).
- **POST `/sync/push`**
 - **Popis:** Zajišťuje nahrání lokálních změn z klientského zařízení na server.
 - **Vstupní parametry (Body):** JSON objekt `SyncPushDto` obsahující pole entit k uložení nebo aktualizaci (`verses`, `exercises`, `friendships`, `user`).
 - **Návratová hodnota:** Objekt potvrzující úspěšné zpracování požadavku: `{ "success": true }`.

6.3.2 Uživatelské endpointy (UserController)

Tento kontroler spravuje operace vázané na uživatelský profil a sociální interakce.

- **POST /user/login**

- **Popis:** Slouží k autentizaci uživatele a inicializaci jeho profilu v databázi. Pokud uživatel v systému neexistuje, je mu automaticky vytvořen účet a přidělen unikátní kód přátelství (`friendCode`).
- **Zabezpečení:** Chráněno pomocí `AuthGuard('jwt-login')`.
- **Návratová hodnota:** Kompletní objekt uživatele (`User`).

- **GET /user/lookup**

- **Popis:** Endpoint pro vyhledávání uživatelů za účelem navázání přátelství na základě unikátního kódu.
- **Vstupní parametry (Query):** `friendCode` (String, šestimístný alfanumerický identifikátor).
- **Návratová hodnota:** Veřejný profil uživatele obsahující `id`, `firstName` a `lastName`.

- **GET /user/:id/stats**

- **Popis:** Slouží k asynchronnímu výpočtu a zobrazení komplexních uživatelských statistik.
- **Vstupní parametry (Param):** `id` (identifikátor uživatele).
- **Návratová hodnota:** Objekt statistik zahrnující `streak` (aktuální denní řada), `totalPractices` (celkový počet procvičení), `memorizedVerses` (počet zapamatovaných veršů na základě úspěšnosti), `timeSpentSeconds` (celkový čas strávený učením) a `score` (bodové hodnocení aktivity).

- **DELETE /user/me**

- **Popis:** Zajišťuje kompletní a bezpečné odstranění uživatelského účtu včetně všech navázaných dat (veršů, procvičení i přátelství) v rámci jedné transakce

pro zachování integrity databáze.

– **Návratová hodnota:** { "success": true }.

6.3.3 Systémové endpointy (AppController)

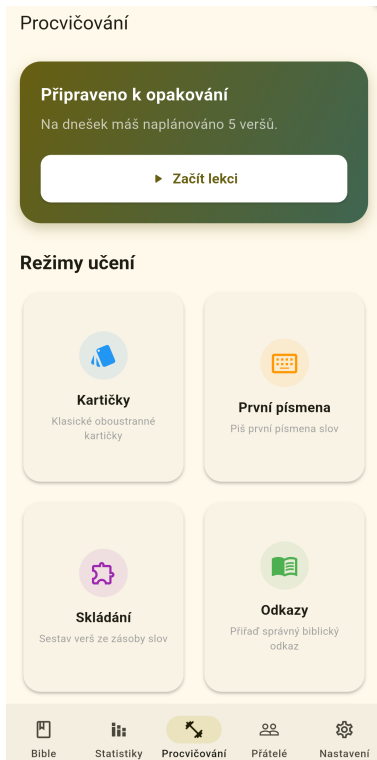
- **GET /health**

– **Popis:** Standardní endpoint pro monitorování dostupnosti a životního cyklu aplikačního kontejneru.

– **Návratová hodnota:** { "status": "ok"}.

6.4 Uživatelské rozhraní

Vzhledem ke zjištěním z úvodní rešerše byl při návrhu klientské části kladen extrémní důraz na použitelnost, přístupnost a dlouhodobou motivaci uživatele. Uživatelské rozhraní (UI) bylo navrženo tak, aby bylo intuitivní a minimalizovalo kognitivní zátěž při každodenním používání. Hlavní rozvržení prvků aplikace je patrné z **obrázku 4**.

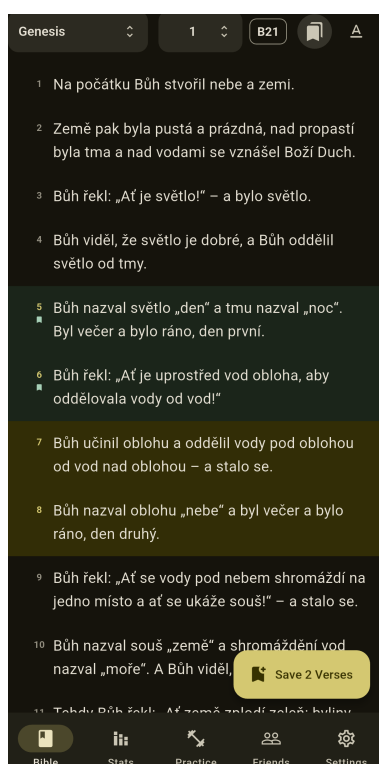


Obrázek 4: Hlavní obrazovka aplikace

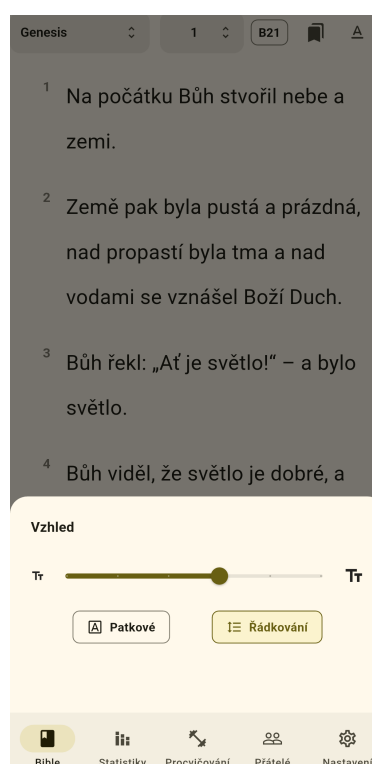
6.4.1 Přístupnost a personalizace

Aby aplikace vyhovovala co nejširšímu spektru uživatelů, obsahuje systém několik úrovní personalizace (viz **obrázky 5 a 6**):

- **Vizuální motivy:** Systém nativně podporuje přepínání mezi světlým (light mode) a tmavým (dark mode) motivem. Tmavý motiv snižuje únavu očí při používání aplikace ve zhoršených světelných podmínkách.
- **Velikost typografie:** Aplikace nabízí možnost uživatelsky upravit velikost zobrazovaného textu. Tato funkce je kritická pro zajištění přístupnosti uživatelům se zhoršeným zrakem, čímž je naplněn jeden z hlavních nefunkčních požadavků.
- **Lokalizace:** Prostředí aplikace je plně lokalizováno. Uživatel si může v nastavení zvolit preferovaný jazyk uživatelského rozhraní, což odstraňuje bariéry pro neanglicky mluvící uživatele.



Obrázek 5: Tmavý motiv (Dark mode)

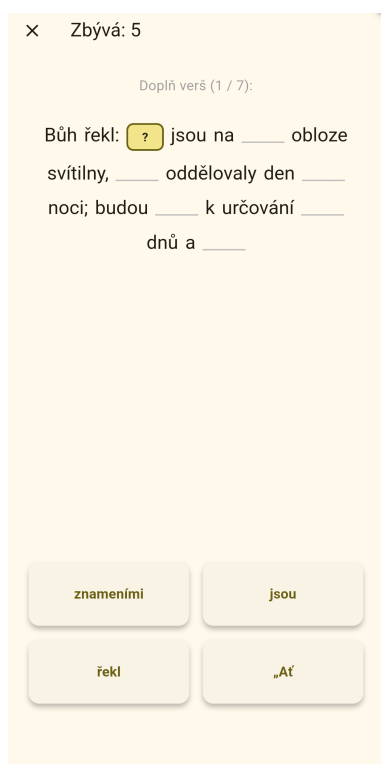


Obrázek 6: Přizpůsobení velikosti textu

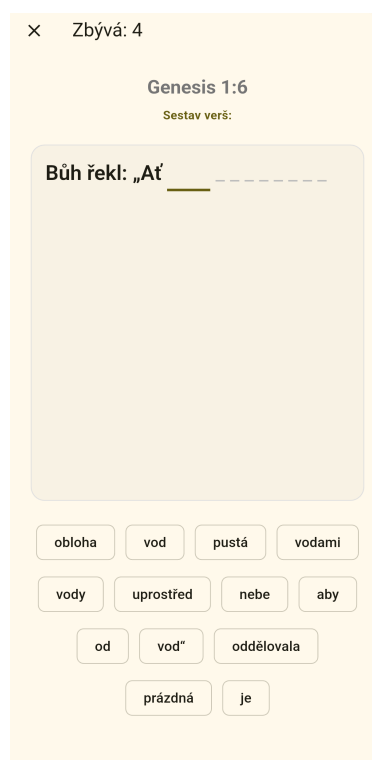
6.5 Gamifikace a motivační prvky

Samotný proces memorizace textu je kognitivně náročný. Pro zvýšení retence (udržení uživatele v aplikaci) byly implementovány prvky gamifikace, které proces učení transformují do poutavější formy:

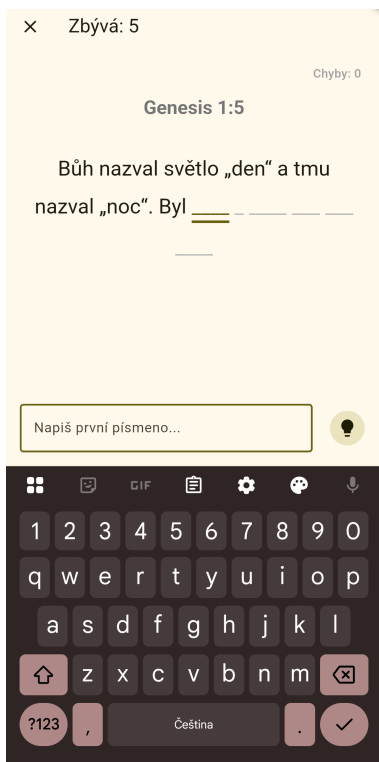
- **Minihry pro učení:** Namísto pasivního čtení je procvičování realizováno formou interaktivních miniher, jako je doplňování slov nebo skládání frází (viz **obrázky 7, 8, 9 a 10**). Tento přístup nutí uživatele nad textem aktivně přemýšlet.



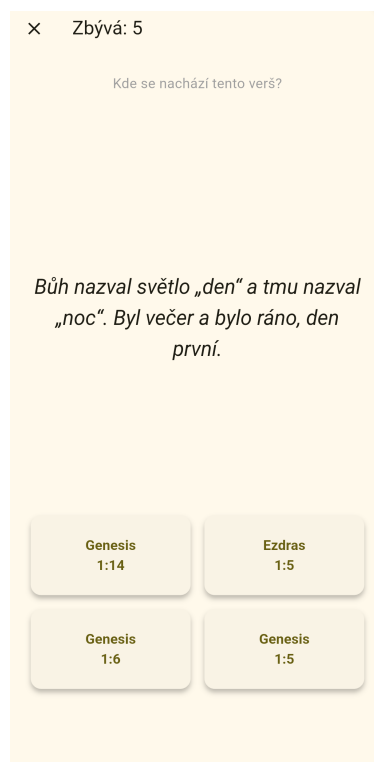
Obrázek 7: Minihra: Výběr slov



Obrázek 8: Minihra: Skládání veršů

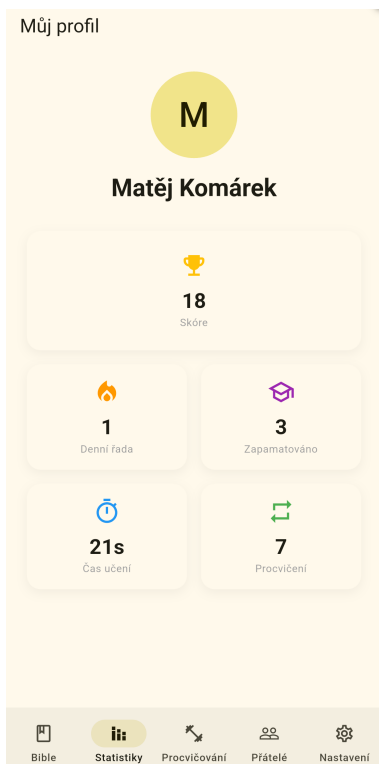


Obrázek 9: Minihra: První písmena

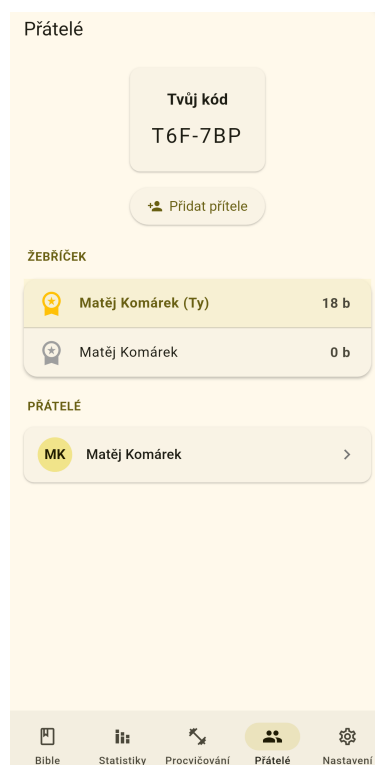


Obrázek 10: Minihra: Reference

- **Sledování denní řady (Streak):** Aplikace vizualizuje konzistenci uživatele, což silně motivuje k pravidelnému návratu.
- **Sociální interakce a žebříčky:** Uživatelé mohou porovnávat své výsledky s přáteli v žebříčcích založených na reálné aktivitě (viz **obrázky 11 a 12**).



Obrázek 11: Osobní statistiky a denní řada (Streak)



Obrázek 12: Žebříček přátel (Leaderboard)

7 Shrnutí výsledků

V rámci realizace maturitního projektu byla úspěšně navržena a implementována klientská mobilní aplikace i serverová část. Výsledný systém Bible Memorizer v aktuálním stavu plně pokrývá veškeré funkční požadavky, přičemž klade mimořádný důraz na prvky gamifikace, které zvyšují uživatelskou retenci a efektivitu učení.

Mezi klíčové dosažené milníky patří:

- **Implementace SM-2 algoritmu s herní smyčkou:** Jádrem aplikace je algoritmus intervalového opakování (Spaced Repetition), který je úzce propojen s herními mechanikami. Úspěšnost uživatele v minihrách (hodnocení 1–5) přímo ovlivňuje obtížnost verše. Tento systém vytváří tzv. *feedback loop*, kde je uživatel motivován k lepším výsledkům, aby mohl „postupovat“ v procesu zapamatování.
- **Systém denních řad (Streaks):** Pro posílení dlouhodobého návyku byl implementován mechanismus *streak*, který vizuálně odměňuje uživatele za každodenní aktivitu. Tento psychologický prvek využívá principu averze ke ztrátě, kdy uživatel usiluje o udržení nepřerušené řady procvičování. Tento systém je podpořen notifikacemi, které se uživateli zobrazí, nepochvilil-li daný den žádný verš.
- **Sociální dynamika a žebříčky:** Aplikace obsahuje pokročilý sociální modul. Uživatelé mohou pomocí unikátních kódů navazovat přátelství a porovnávat své statistiky a skóre v žebříčcích. Tento prvek zdravé soutěživosti transformuje individuální učení v kolektivní zážitek.
- **Offline režim a synchronizace:** Aby nebyla herní smyčka přerušena nedostatkem konektivity, byl vyvinut synchronizační mechanismus využívající lokální databázi SQLite a serverový PostgreSQL. Postup uživatele, jeho skóre i dosažené úspěchy jsou bezpečně ukládány a při obnovení připojení automaticky synchronizovány pomocí časových razítek (*timestamps*).

Aktuálně je aplikace ve stavu plně funkčního a otestovaného prototypu, který efektivně kombinuje vzdělávací cíle s gamifikací.

Aplikace je dostupná ke stažení na následujícím odkazu:

<https://b2022komama.delta-www.cz/>

Závěr

Lze konstatovat, že všechny hlavní cíle projektu byly naplněny. Vytvořená platforma Bible Memorizer demonstruje funkční propojení frameworků Flutter a NestJS s vědeckými metodami učení. Díky implementaci algoritmu intervalového opakování (SM-2) a prvků gamifikace aplikace efektivně zvyšuje motivaci uživatelů a řeší kritické nedostatky dosavadní konkurence.

Ve srovnání s analyzovanými řešeními (VerseLocker, Bible Memory, Remember Me a BibleMe) přináší aplikace zásadní inovace, zejména integrovanou čtečku a spravedlivý sociální systém. Ten na rozdíl od aplikace VerseLocker nehodnotí subjektivní odhad zapamatování, ale skutečnou aktivitu v minihrách a délku denní řady. Výsledky srovnání shrnuje Tabulka 1.

| Aplikace | Int. opakov. | Čtečka Bible | Žebříček | České UI |
|------------------------|--------------|--------------|--------------------------|------------|
| Bible Memorizer | Ano | Ano | Ano (Spravedlivý) | Ano |
| Bible Memory | Ano | Ano | Ne | Ne |
| VerseLocker | Ne | Ne | Ano* | Ne |
| Remember Me | Ne | Ne | Ne | Ne** |
| BibleMe | Ne | Ne | Ne | Ne |

Tabulka 1: Srovnání funkcionality aplikace s existujícími řešeními

* Žebříček hodnotí pouze počet veršů bez testování znalostí.

** Podporuje český překlad Bible, ale rozhraní aplikace nikoliv.

Největší technologickou výzvou byla obousměrná synchronizace dat mezi SQLite a PostgreSQL. Problém potenciálních konfliktů při úpravách ve více zařízeních v offline režimu byl vyřešen mechanismem založeným na časových razítkách (timestamps), který zajišťuje, že systém jako platný vždy přijímá nejnovější záznam.

Výsledkem je moderní, bezpečná a plně funkční platforma optimalizovaná pro Android. Ačkoliv je aplikace ve stavu funkčního prototypu, existuje prostor pro další vývoj. Hlavní možností pro rozvoj je nahrazení algoritmu SM-2 pokročilejším modelem HLR (Half-Life Regression) [15], který dokáže na základě historie paměťových stop přesněji predikovat pravděpodobnost zapomení a dále tak optimalizovat proces učení.

Reference

- [1] Icez, *Křivka zapomínání*. Online. In: WIKIMEDIA COMMONS. 2025. Dostupné z: <https://commons.wikimedia.org/wiki/File:ForgettingCurve.svg>. [cit. 2026-03-25].
- [2] HACKETT, Conrad. *How the Global Religious Landscape Changed From 2010 to 2020*. Online. Pew Research Center. Dostupné z: <https://www.pewresearch.org/religion/2025/06/09/how-the-global-religious-landscape-changed-from-2010-to-2020/>. [cit. 2026-03-26].
- [3] EBBINGHAUS, Hermann. *Memory: a contribution to experimental psychology*. Nabu Press, 2012. ISBN 978-1272622725.
- [4] SCRIPTURE MEMORY FELLOWSHIP. *Bible Memory: VerseLocker*. Online. 2018, aktualizováno 16.7.2025. Dostupné z: <https://play.google.com/store/apps/details?id=com.audioverse.smf.audioverse>. [cit. 2026-03-26].
- [5] MILLENNIAL APPS - THE BIBLE MEMORY APP. *The Bible Memory App*. Online. 2013, aktualizováno 13.3.2026. Dostupné z: <https://play.google.com/store/apps/details?id=com.millennialsolutions.scripturetyper>. [cit. 2026-03-26].
- [6] REMEMBER ME - BIBLE MEMORY APP. POIMENA. *Bible Memory*. Online. 2010, aktualizováno 22.3.2026. Dostupné z: https://play.google.com/store/apps/details?id=org.bible.remember_me. [cit. 2026-03-26].
- [7] EFIRE APPS. Bible Memory: *BibleMe*. Online. 2016, aktualizováno 16.3.2026. Dostupné z: <https://play.google.com/store/apps/details?id=com.efireapps.bibleme>. [cit. 2026-03-26].
- [8] PUBMED CENTRAL. *The right time to learn: mechanisms and optimization of spaced learning*. Online. 2016. Dostupné z: <https://pmc.ncbi.nlm.nih.gov/articles/PMC5126970/>. [cit. 2026-03-26].
- [9] *Flutter documentation*. Online. 2017, aktualizováno 9.3.2026. Dostupné z: <https://docs.flutter.dev/>. [cit. 2026-03-26].

- [10] *NestJS documentation*. Online. 2017. Dostupné z: <https://docs.nestjs.com/>. [cit. 2026-03-26].
- [11] *PostgreSQL 18.3 Documentation*. Online. 2025. Dostupné z: <https://www.postgresql.org/docs/18/index.html>. [cit. 2026-03-26].
- [12] *SQLite Documentation*. Online. 2000, aktualizováno 13.11.2025. Dostupné z: <https://sqlite.org/docs.html>. [cit. 2026-03-26].
- [13] *Cloudflare Docs*. Online. 2010. Dostupné z: <https://developers.cloudflare.com/>. [cit. 2026-03-26].
- [14] SUPERMEMO. *Application of a computer to improve the results obtained in working with the SuperMemo method*. Online. 1990. Dostupné z: <https://www.supermemo.com/en/blog/application-of-a-computer-to-improve-the-results-obtained-in-working-with-the-supermemo-method>. [cit. 2026-03-26].
- [15] SETTLES, Burr a MEEDER, Brendan. *A Trainable Spaced Repetition Model for Language Learning*. Online. 2016. Dostupné z: <https://research.duolingo.com/papers/settles.acl16.pdf>. [cit. 2026-03-26].

Seznam obrázků

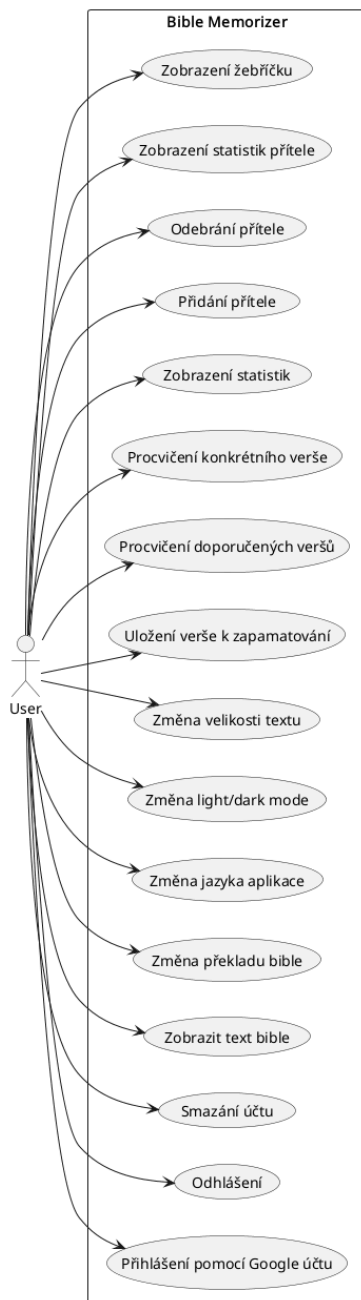
| | | |
|----|--|----|
| 1 | Křivka zapomínání [1] | 10 |
| 2 | Architektura projektu | 16 |
| 3 | Datový model | 18 |
| 4 | Hlavní obrazovka aplikace | 23 |
| 5 | Tmavý motiv (Dark mode) | 24 |
| 6 | Přizpůsobení velikosti textu | 24 |
| 7 | Minihra: Výběr slov | 25 |
| 8 | Minihra: Skládání veršů | 25 |
| 9 | Minihra: První písmena | 26 |
| 10 | Minihra: Reference | 26 |
| 11 | Osobní statistiky a denní řada (Streak) | 27 |
| 12 | Žebříček přátel (Leaderboard) | 27 |
| 13 | Use Case diagram aplikace | 34 |
| 14 | Detailní architektura tříd systému Bible Memorizer | 35 |

Seznam příloh

- **Příloha 1:** Use Case diagram a specifikace případů užití
- **Příloha 2:** UML Class diagram (Diagram tříd) backendu

Přílohy

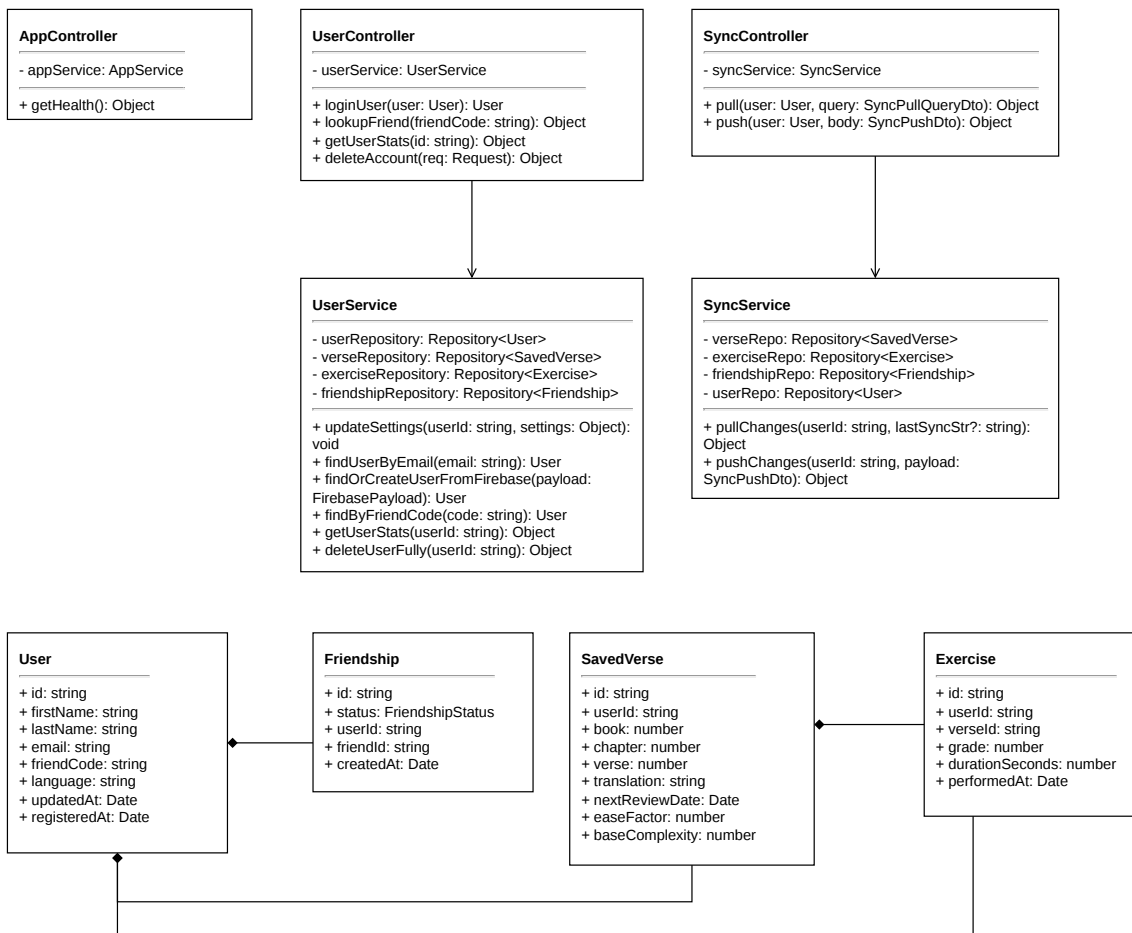
Příloha 1: Use Case diagram a specifikace případů užití



Obrázek 13: Use Case diagram aplikace

Příloha 2: UML Class diagram backendu

V této příloze je zobrazen detailní diagram tříd serverové části aplikace (NestJS). Diagram vizualizuje vnitřní strukturu kontrolerů, servisů a entit včetně jejich vzájemných vazeb a metod.



Obrázek 14: Detailní architektura tříd systému Bible Memorizer