

# Dokumentace k projektu Šibenice

1. března 2025

## 1 Úvod

Tento dokument slouží jako technická dokumentace k projektu **Šibenice**, což je mobilní hra určená pro platformy **Android** a **iOS**. Hra je postavena na technologii **React Native** s využitím frameworku **Expo** a poskytuje uživatelsky přívětivé rozhraní pro procvičování slovní zásoby z různých kategorií.

Hlavním cílem aplikace je vytvořit interaktivní a edukativní herní prostředí, které uživateli umožní hravou formou testovat a rozvíjet slovní zásobu. Hra zahrnuje různé obtížnosti, skórovací systém a možnosti přizpůsobení.

Tato dokumentace pokrývá následující oblasti:

- Architektura a technologie použité v aplikaci,
- Popis implementace a struktury kódu,
- Instrukce ke spuštění aplikace,
- Přehled funkcionalit a jejich využití,
- Ukázky obrazovek a jejich popis.

Cílem dokumentace je poskytnout podrobný přehled o vývoji aplikace a umožnit její rozšiřování nebo úpravy v budoucnu.

## 2 Architektura aplikace

Aplikace **Šibenice** je postavena na technologii **React Native** a využívá framework **Expo**, který zjednodušuje vývoj, testování a nasazení mobilních aplikací. Struktura projektu je navržena modulárně, aby byla umožněna snadná údržba a rozšiřitelnost funkcionalit.

### 2.1 Modulární přístup

Projekt je rozdělen do jednotlivých komponent, které pokrývají různé funkční části aplikace. Mezi klíčové moduly patří:

- **Hlavní obrazovka (Menu)** – výběr tématu a obtížnosti,
- **Obrazovka s úrovněmi (Levels)** – seznam dostupných úrovní pro zvolené téma a obtížnost,

- **Herní obrazovka (Game)** – samotný průběh hry, zobrazení šibenice a hádání písmen,
- **Výherní obrazovka (Win)** – animace a skórování po úspěšném dokončení úrovně,
- **Nastavení (Settings)** – možnosti přizpůsobení hry, včetně zvuků, tématu a obtížnosti.

## 2.2 Navigace v aplikaci

Aplikace využívá systém navigace **Expo Router**, který umožňuje přehlednou správu jednotlivých obrazovek a přechod mezi nimi. Navigace je založena na hierarchickém uspořádání stránek:

- `index.tsx` – hlavní stránka aplikace,
- `menu.tsx` – přechod na výběr tématu a obtížnosti,
- `levels.tsx` – seznam úrovní pro konkrétní téma a obtížnost,
- `game.tsx` – samotná hra,
- `win.tsx` – obrazovka s oznámením o výhře,
- `settings.tsx` – nastavení aplikace.

Každá z těchto obrazovek je samostatná komponenta, která komunikuje s ostatními částmi aplikace pomocí parametrů v navigaci.

## 2.3 Ukládání dat

Aplikace využívá **AsyncStorage** pro uchování uživatelských preferencí a skóre. Hlavní uložené informace zahrnují:

- **Nastavení uživatele** – barevné téma, zvuky, vibrace, časový limit.
- **Skóre** – uložené nejlepší výsledky pro jednotlivá slova a úrovně.

## 2.4 Použité knihovny

Aplikace se opírá o několik knihoven pro usnadnění vývoje a vylepšení uživatelského rozhraní:

- `react-native-vector-icons` – ikony pro vizuální prvky,
- `react-native-progress` – ukazatel průběhu ve hře,
- `react-native-confetti-cannon` – animace oslav při výhře,
- `expo-router` – správa navigace mezi obrazovkami.

## 3 Hlavní funkcionality aplikace

Aplikace **Šibenice** obsahuje několik klíčových funkcionalit, které umožňují interaktivní a dynamickou hratelnost. Tyto funkcionality zahrnují výběr témat a obtížností, správu skóre, časový limit a vizuální zpětnou vazbu.

### 3.1 Výběr tématu a obtížnosti

Při spuštění hry si uživatel nejprve zvolí herní téma a úroveň obtížnosti. Témata jsou rozdělena do několika kategorií, jako například:

- **Zvířata**,
- **Cestování**,
- **Jídlo**,
- **Auta**,
- **Technologie**,
- **Filmy**,
- **Historie**,
- **Věda** a další.

Obtížnosti jsou rozděleny na tři úrovně:

- **Začátečník** – jednodušší slova s více nápovědami,
- **Pokročilý** – střední úroveň obtížnosti s méně nápovědami,
- **Expert** – nejtěžší úroveň, často se složitějšími slovy.

Výběr obtížnosti ovlivňuje herní mechaniku, například počet pokusů na uhodnutí slova.

### 3.2 Průběh hry a pravidla

Hráč hádá slovo zadáváním jednotlivých písmen. Pokud se písmeno nachází ve slově, je odhaleno na správných pozicích. Pokud ne, je hráči odečten jeden pokus. Hra je ukončena v těchto případech:

- Hráč správně uhodne celé slovo – zobrazí se animace výhry a přidělené skóre.
- Hráč vyčerpá maximální počet nesprávných pokusů – zobrazí se obrazovka s neúspěchem.
- Vyprší časový limit (pokud je aktivován) – hráč automaticky prohrává kolo.

### 3.3 Skórovací systém

Každé slovo lze ohodnotit na základě počtu chyb. Skórování probíhá podle těchto pravidel:

- **3 hvězdy** – žádná chyba,
- **2 hvězdy** – maximálně 2 chyby,
- **1 hvězda** – maximálně 4 chyby,
- **0 hvězd** – více než 4 chyby.

Skóre se ukládá pomocí **AsyncStorage** a při opětovném hraní stejné úrovně se započítá nejvyšší dosažené skóre.

### 3.4 Nápořědy

Pro usnadnění hry je k dispozici nápořěda, která poskytuje krátký popis hádaného slova. Nápořědu lze zobrazit kliknutím na odpovřídající tlačřtko.

### 3.5 Časový limit a pokusy

Hra podporuje možnost nastavení časového limitu pro dokončení úrovně. Pokud je aktivován, odpočřtává se čas a při jeho vypršení hra automaticky končí prohrou. Dále lze aktivovat omezený počet pokusů na hádání slova.

### 3.6 Animace a vizuální efekty

Aplikace využívá různé vizuální efekty pro zvýšení interaktivity:

- Animace konfet při výhře (**ConfettiCannon**),
- Ukazatel průběhu (**react-native-progress**),
- Vibrace zařízení při nesprávné odpovředi (pokud je aktivována).

## 4 Uživatelské rozhraní a navigace

Aplikace **Šibenice** využívá moderní a přehledné uživatelské rozhraní založené na komponentách **React Native**. Rozhraní je optimalizováno pro mobilní zařízení s různými velikostmi obrazovek a podporuje přepřínání mezi světlým a tmavým režimem.

### 4.1 Hlavní obrazovky aplikace

Aplikace se skládá z několika hlavních obrazovek, mezi kterými se uživatel může pohybovat prostřednictvím navigačního systému **Expo Router**:

- **Úvodní obrazovka (`index.tsx`)** – zobrazuje logo aplikace a základní možnosti (spuštění hry, nastavení).
- **Hlavní menu (`menu.tsx`)** – umožňuje výběr tématu a obtížnosti.

- **Seznam úrovní (levels.tsx)** – zobrazuje dostupné úrovně s vizuální indikací dosaženého skóre.
- **Herní obrazovka (game.tsx)** – obsahuje interaktivní herní plochu s možností hádání slov.
- **Obrazovka výhry (win.tsx)** – zobrazí se po úspěšném dokončení úrovně.
- **Obrazovka nastavení (settings.tsx)** – umožňuje změnit téma, zvuk, vibrace a další nastavení.

## 4.2 Hlavní menu

Po spuštění aplikace se uživateli zobrazí hlavní menu, kde si může vybrat téma hry a úroveň obtížnosti. Menu je přehledně rozvržené do kategorií, které jsou barevně odlišeny:

- Každé téma je reprezentováno ikonou a barvou (např. **Zvířata** – oranžová, **Technologie** – modrá).
- Obtížnosti mají vlastní barevné kódování: **Začátečník** (modrá), **Pokročilý** (oranžová), **Expert** (červená).

## 4.3 Herní obrazovka

Herní obrazovka je navržena tak, aby byla co nejintuitivnější:

- Zobrazuje aktuální slovo jako sérii podtržítok, kde se postupně doplňují správně uhodnutá písmena.
- Pod hádaným slovem je abecední klávesnice, která umožňuje výběr písmen.
- Indikátor pokusů (v případě omezených pokusů) nebo odpočet času (pokud je aktivován časový limit).
- Možnost zobrazení nápovědy.

Při správném uhodnutí slova se zobrazí animace výhry a přesměrování na obrazovku s výsledkem.

## 4.4 Obrazovka výsledku

Po dokončení úrovně se uživateli zobrazí stránka s výsledkem:

- Počet hvězd (hodnocení dle počtu chyb),
- Možnost pokračovat na další úroveň nebo se vrátit do menu,
- Krátká animace konfet pro zvýraznění výhry.

## 4.5 Navigace mezi obrazovkami

Navigace mezi obrazovkami probíhá prostřednictvím **Expo Router**, který umožňuje přehledné a efektivní řízení přechodů. Každá stránka aplikace je reprezentována samostatnou komponentou a mezi stránkami se přechází pomocí parametrů v URL.

Uživatel může navigovat:

- **Tlačítka zpět** – návrat na předchozí stránku (např. zpět do menu),
- **Automatickými přechody** – po dokončení úrovně je hráč přesměrován na výsledkovou obrazovku,
- **Přímým výběrem** – uživatel si může kdykoliv vybrat jinou úroveň v seznamu úrovní.

## 4.6 Přizpůsobení vzhledu

Aplikace podporuje přepínání mezi světlým a tmavým režimem, které se nastavuje v sekci **Nastavení**. Volba se ukládá pomocí **AsyncStorage** a při dalším spuštění se automaticky obnoví.

# 5 Technická implementace a struktura kódu

Aplikace **Šibenice** je postavena na moderním technologickém stacku s využitím **React Native** a **Expo**. Kód je rozdělen do několika modulů, což umožňuje snadnou údržbu a rozšiřitelnost projektu.

## 5.1 Adresářová struktura

Projekt je organizován podle doporučených postupů pro aplikace v **Expo Router**. Hlavní složky jsou následující:

```
/
|-- assets/           % Obsahuje fonty, obrázky a zvuky
|-- components/       % Opakovaně použitelné React komponenty
|-- screens/          % Hlavní obrazovky aplikace
|   |-- Game.tsx      % Herní obrazovka
|   |-- Menu.tsx      % Výběr tématu a obtížnosti
|   |-- Levels.tsx    % Seznam úrovní
|   |-- Win.tsx       % Obrazovka výhry
|   |-- Settings.tsx  % Nastavení hry
|   |-- Index.tsx     % Úvodní obrazovka aplikace
|-- data/             % JSON soubory se slovy a nastavením
|   |-- words.json    % Slova podle témat a obtížností
|   |-- buttons.json  % Konfigurace tlačítek pro menu
|-- utils/            % Pomocné funkce a utility
|-- App.tsx           % Hlavní soubor aplikace
|-- package.json      % Konfigurace balíčků
```

Tato struktura umožňuje jasné oddělení jednotlivých částí aplikace.

## 5.2 Klíčové komponenty

Každá část aplikace je implementována jako samostatná komponenta v souboru `.tsx`. Mezi nejdůležitější komponenty patří:

### 5.2.1 `Game.tsx` – herní logika

- Obsahuje hlavní herní mechanismus.
- Načítá slova z JSON souboru podle zvoleného tématu a obtížnosti.
- Sleduje správné a nesprávné odpovědi.
- Zobrazuje šibenici a progres hráče.
- Při výhře nebo prohře přesměrovává hráče na odpovídající obrazovku.

### 5.2.2 `Menu.tsx` – výběr tématu a obtížnosti

- Obsahuje seznam témat a obtížností, které jsou načítány z JSON souboru.
- Při výběru se parametry předávají na stránku `Levels.tsx`.
- Každé téma a obtížnost mají unikátní barvu a ikonu.

### 5.2.3 `Levels.tsx` – seznam úrovní

- Načítá seznam úrovní na základě zvoleného tématu a obtížnosti.
- Indikuje dosažené skóre pomocí hvězdiček.
- Přesměrovává hráče na `Game.tsx` s odpovídající úrovní.

### 5.2.4 `Win.tsx` – obrazovka vítězství

- Zobrazuje hráči dosažené skóre a efekt konfet.
- Poskytuje možnosti pokračování na další úroveň nebo návrat do menu.

### 5.2.5 `Settings.tsx` – nastavení hry

- Umožňuje změnu tématu aplikace (tmavý/světlý režim).
- Nabízí možnost aktivace/deaktivace zvuků a vibrací.
- Umožňuje nastavení časového limitu pro hru.
- Všechna nastavení se ukládají pomocí `AsyncStorage`.

## 5.3 Práce s daty

Aplikace využívá soubory JSON pro uchování herních dat:

- **words.json** – obsahuje slova, jejich kategorie a nápovědy.
- **buttons.json** – obsahuje seznam obtížností a témat s barvami a ikonami.
- **AsyncStorage** – ukládá skóre hráče a uživatelská nastavení.

## 5.4 Příklad práce s AsyncStorage

Kód pro načtení a uložení skóre hráče:

```
const loadScores = async () => {
  try {
    const savedScores = await AsyncStorage.getItem("score");
    if (savedScores) {
      setScoreData(JSON.parse(savedScores));
    }
  } catch (error) {
    console.error("Chyba při načítání skóre:", error);
  }
};

const saveScore = async (word: string, newScore: number) => {
  try {
    const oldScore = scoreData[word] ?? 0;
    const betterScore = Math.max(oldScore, newScore);
    const updatedScoreData = { ...scoreData, [word]: betterScore };

    await AsyncStorage.setItem("score", JSON.stringify(updatedScoreData));
    setScoreData(updatedScoreData);
  } catch (error) {
    console.error("Chyba při ukládání skóre:", error);
  }
};
```

Tento mechanismus umožňuje uložení nejlepšího skóre pro jednotlivá slova.

# 6 Testování, ladění a nasazení aplikace

V této kapitole jsou popsány způsoby testování a ladění aplikace během vývoje a postupy potřebné pro její nasazení na mobilní platformy Android a iOS.

## 6.1 Testování aplikace

Testování aplikace probíhá v několika fázích:

- **Vývojové testování** – průběžné testování funkcionalit během vývoje.



- **Automatizované testy** – kontrola správnosti funkcí pomocí testovacích knihoven.
- **Testování na fyzických zařízeních** – ověření chování aplikace v reálných podmínkách.

### 6.1.1 Vývojové testování v Expo

Během vývoje je aplikace spouštěna v Expo Developer Tools pomocí příkazu:

```
npx expo start
```

Tento nástroj umožňuje:

- Spouštění aplikace v emulátorech i na fyzických zařízeních (Android/iOS).
- Automatický reload při změnách kódu (*Fast Refresh*).
- Přístup ke konzoli pro ladění chyb v kódu.

### 6.1.2 Automatizované testy

Aplikace může využívat testovací frameworky jako:

- **Jest** – testování jednotlivých funkcí aplikace.
- **React Native Testing Library** – simulace interakce s uživatelským rozhraním.

Příklad základního testu pro ověření existence komponenty:

```
import { render } from "@testing-library/react-native";
import Game from "../screens/Game";

test("Zkontroluje renderování herní obrazovky", () => {
  const { getByText } = render(<Game />);
  expect(getByText("Hádej slovo")).toBeTruthy();
});
```

### 6.1.3 Testování na fyzických zařízeních

Testování na reálných mobilních zařízeních se provádí pomocí:

- **Expo Go** – umožňuje spustit aplikaci na mobilu bez nutnosti jejího sestavení.
- **Android Emulator / iOS Simulator** – simulace aplikace v prostředí dané platformy.
- **TestFlight** – distribuce testovacích verzí aplikace pro iOS.

## 6.2 Ladění aplikace

Expo poskytuje vestavěné nástroje pro ladění:

- **React Developer Tools** – umožňuje inspekci komponent v reálném čase.
- **Debugging Console** – pro zobrazení chyb a výstupů z konzole.
- **Remote Debugging** – možnost ladění kódu přímo v prohlížeči.

### 6.2.1 Ladění chyb pomocí výpisů do konzole

V případě chyb lze použít `console.log()` pro diagnostiku:

```
useEffect(() => {  
  console.log("Aktuální téma:", topic);  
}, [topic]);
```

Další možnosti ladění zahrnují použití breakpoints v Chrome DevTools.

## 6.3 Nasazení aplikace

Aplikace může být nasazena dvěma způsoby:

- **Expo Managed Workflow** – snadné sestavení a distribuce pomocí Expo Cloud Build.
- **Eject do Bare Workflow** – přechod na nativní vývoj s možností integrace nativních modulů.

### 6.3.1 Příprava na produkční sestavení

Pro nasazení je nutné vytvořit optimalizovanou verzi aplikace:

```
eas build --platform android  
eas build --platform ios
```

Tento proces vygeneruje instalační soubory `.apk` nebo `.aab` pro Android a `.ipa` pro iOS.

### 6.3.2 Publikace na Google Play a App Store

Publikace aplikace probíhá v několika krocích:

1. Registrace vývojářského účtu na **Google Play Console** nebo **Apple Developer Program**.
2. Nahrání sestavené aplikace do příslušného obchodu.
3. Vyplnění metadat (popis, obrázky, kategorie).
4. Odeslání k recenzi a schválení.

Pro usnadnění nasazení lze využít příkaz:

```
eas submit --platform android
```

## 7 Budoucí vylepšení a možné rozšíření

Aplikace **Šibenice** poskytuje stabilní a dobře strukturovaný základ, který umožňuje její další rozšiřování. Tato kapitola se zabývá návrhy možných vylepšení, která mohou zvýšit uživatelský komfort a rozšířit funkcionalitu aplikace.

## 7.1 Možná rozšíření funkcionality

### 7.1.1 Online multiplayer mód

Jednou z nejzajímavějších možností rozšíření je implementace **multiplayer režimu**, kde by hráči mohli hrát proti sobě v reálném čase. Tento režim by mohl zahrnovat:

- Připojení k online serveru a možnost pozvat přátele ke hře.
- Systém bodování a žebříčku nejlepších hráčů.
- Možnost hraní proti náhodným soupeřům.

Technologicky by multiplayer mohl být postaven na **WebSockets** nebo **Firestore Realtime Database** pro synchronizaci hry mezi hráči.

### 7.1.2 Personalizovaný tréninkový režim

Aplikace by mohla obsahovat **tréninkový režim**, kde by hráči mohli procvičovat konkrétní slova nebo tematické okruhy dle vlastního výběru. Možné funkce:

- Výběr obtížnosti na základě historie hráče.
- Statistické vyhodnocení pokroků ve slovní zásobě.
- Možnost opakování slov, která hráč neuhodl.

### 7.1.3 Vylepšený systém nápovědy

Současná nápověda poskytuje pouze textový popis slova. Možná vylepšení:

- Zobrazení obrázkové nápovědy (přidání obrázků k jednotlivým slovům).
- Možnost odhalení prvního nebo posledního písmene jako součást nápovědy.
- Systém odemykatelných nápověd na základě nasbíraného skóre.

### 7.1.4 Tématické události a výzvy

Aplikace by mohla nabízet **časově omezené výzvy**, které by motivovaly hráče k pravidelnému hraní. Možné implementace:

- Speciální týdenní/měsíční výzvy se zaměřením na určitá témata.
- Sezónní události (např. vánoční, letní edice s novými slovy).
- Odměny za dokončení výzev, například nové vizuální motivy aplikace.

## 7.2 Vylepšení uživatelského rozhraní a designu

### 7.2.1 Možnosti personalizace vzhledu

Aplikace by mohla umožnit větší **personalizaci vzhledu** podle preferencí uživatele:

- Možnost výběru barevného schématu aplikace.
- Různé vizuální styly šibenice.
- Možnost změny fontu a velikosti písma.

### 7.2.2 Animace a efekty

Zlepšení vizuálních efektů by mohlo aplikaci zpříjemnit pro hráče:

- Plynulé animace při zadávání písmen.
- Efekty zvýraznění při správném/špatném odhadu.
- Animovaný progres indikující postup ve hře.

## 7.3 Lepší integrace s externími službami

### 7.3.1 Ukládání progresu na cloud

Možná implementace ukládání dat do cloudu, aby hráč nepřišel o svůj postup:

- Integrace s **Firebase** nebo jinou cloudovou službou pro synchronizaci dat mezi zařízeními.
- Přihlášení pomocí Google/Facebook pro zálohování progresu.

### 7.3.2 Podpora více jazyků

Aplikace by mohla být rozšířena o podporu více jazyků:

- Možnost přepnutí mezi různými jazyky hry.
- Slovníky přizpůsobené pro různé jazykové mutace.

## 7.4 Optimalizace výkonu

Pro lepší výkon aplikace by bylo možné:

- Zredukovat velikost balíčku odstraněním nepotřebných knihoven.
- Optimalizovat načítání slov a assetů pro rychlejší odezvu aplikace.
- Použití **Hermes Engine** pro efektivnější běh aplikace na Androidu.

## 8 Závěr

Tato dokumentace poskytuje ucelený přehled o vývoji, architektuře, funkcionalitě a možnostech rozšíření mobilní hry **Šibenice**. Aplikace byla navržena jako interaktivní a edukativní hra, která umožňuje hráčům procvičovat slovní zásobu z různých tematických oblastí.

Hlavní přínosy aplikace:

- Podpora multiplatformního vývoje pro **Android** a **iOS**.
- Snadná rozšiřitelnost díky použití **React Native** a **Expo**.
- Intuitivní uživatelské rozhraní a přizpůsobitelná herní mechanika.
- Ukládání herního progresu pomocí **AsyncStorage**.
- Možnost budoucích vylepšení, včetně online multiplayeru, cloudového ukládání a širší personalizace.

## 8.1 Shrnutí vývoje

Během vývoje byly použity moderní technologie, které usnadňují správu kódu a optimalizují výkon aplikace. Klíčové části implementace zahrnovaly:

- Implementaci navigace mezi obrazovkami pomocí **Expo Router**.
- Použití knihovny **react-native-vector-icons** pro vizuální zlepšení.
- Integraci zvukových efektů a animací pro lepší uživatelský zážitek.
- Strukturované zpracování dat uložených ve formátu **JSON**.
- Možnost přepínání mezi světlým a tmavým režimem.

## 8.2 Další kroky

V současné verzi aplikace je základní herní funkcionalita plně funkční. Další vývoj by se mohl zaměřit na:

- Implementaci online režimu pro soutěžení hráčů v reálném čase.
- Rozšíření databáze slov a přidání nových tematických okruhů.
- Vylepšení vizuálních efektů a animací.
- Integraci cloudového ukládání progresu pomocí **Firebase**.

## 8.3 Závěrečné hodnocení

Aplikace **Šibenice** představuje kvalitní základ pro mobilní vzdělávací hru, která kombinuje zábavu s učením. Díky flexibilní architektuře je možné ji dále rozšiřovat a přizpůsobovat novým požadavkům uživatelů. Tento dokument by měl sloužit jako průvodce pro budoucí údržbu a rozvoj projektu.