AI Storybook – Project overview

AI Storybook is a Flutter-based application for children that generates creative stories with the help of artificial intelligence, fully offline. The stories are created using Ollama, an open-source tool that runs large language models (LLMs) locally. Thanks to this, AI story generation works 100% without an internet connection, so user data never leaves the device and there are no cloud service costs. The main screen (HomeScreen) serves as the starting view: from here, users can create a new story or access the local story library. The interface is designed with large, easy-to-read fonts and a child-friendly blue Material 3 theme, making navigation and reading simple and enjoyable for kids.

Main features

- **Local AI story generation:** The user selects a topic, and the application creates a complete story locally with the Ollama LLM, without requiring an internet connection.

- **Page-flip storybook view:** The application displays the story page by page using the Flutter PageView widget.

- **Local library:** The generated stories can be saved to local storage. The saved stories are stored in a searchable library (based on a Hive database), where any story can be opened or deleted. **Hive** is a fast, lightweight NoSQL database for Flutter/Dart that uses key–value pairs and does not require a predefined schema.

- **Child-friendly UI:** The user interface uses bright colors and large buttons; titles are displayed in a 32px bold blue font, while body text uses an 18px normal-weight font. It takes advantage of modern Material 3 design, which provides dynamic color palettes and improved accessibility.


AI Storybook
Üdvözöl az AI Storybook!
Készíts csodálatos meséket AI segítségével
Új mese generálása
Könyvtáram

- **100% offline operation:** All components of the application run locally: the artificial intelligence (Ollama), the images, and the data storage (Hive) all use the device's internal resources. This means there is no need for an internetconnection or external API keys, and all data remains local.

Technolgy background

- **Flutter (Dart**): The project was built with the Flutter framework, which makes it possible to create Android, iOS, web, and desktop applications from the same codebase. Flutter 3.0+ uses Material 3 design by default, which is also applied in this application.
- **Ollama:** An open-source tool for running large language models locally. It enables models such as Llama2 or Llama3 to run completely offline on the user's device.
- **Hive:** A key–value-based local database. In the application, the stories box is used to store Story objects. Hive also supports type encoding: with a custom TypeAdapter, any Dart object can be stored and retrieved.
- **Material 3 and responsive design: U**sing Flutter's ThemeData.colorScheme.fromSeed method, a blue color palette is set, and with the useMaterial3: true flag the latest Material Design specification is followed. The sizing and layout of UI elements ensure that the app remains usable across different screen sizes.

User interface and usage

From the main screen, the user can start generating a new story or enter the library. When requesting a new story, they provide a topic and then tap the "Generate" button to launch the offline AI model. The finished story can be read page by page by swiping left or right; the current page is indicated by dots at the bottom of the screen. Each page includes a colorful illustration (placeholder image) and the corresponding text. Readability is supported by large, high-contrast fonts and a child-friendly color scheme. With the "Save to Library" button, the user can store the current story, which can later be reopened or deleted from the library list. Across the interface, special attention was given to a child-friendly design (e.g., large buttons and intuitive icons).

Architectura & Code

The program starts in the **main.dart** file. In the main() function, the Flutter environment is first initialized (WidgetsFlutterBinding.ensureInitialized()), then the Hive database is set up with Hive.initFlutter(), and the stories box (where the stories are stored) is opened. The StoryAdapter type is also registered with Hive so that Story objects can be saved. After this, runApp(AIStorybookApp()) launches the application's graphical interface.

The AIStorybookApp class's build method creates the MaterialApp: it sets the title ("AI Storybook"), the theme (blue colors with Material 3), and the customized text styles. The

main screen is provided by the HomeScreen component, and the debug banner is hidden (debugShowCheckedModeBanner: false).

In summary, the code is structured so that on startup the Hive database is prepared, the user interface style is configured, and then the first screen is displayed.

Data storage (StorageService)

The **StorageService** class is responsible for saving stories locally using Hive. In its constructor, it opens the stories box with Hive.box<Story>('stories'). In the saveStory(Story story) method, it first checks whether a story with the same title already exists in the box (case-insensitive). If such a story is found, the old one is deleted: it locates the key of the existing record (using firstWhere) and removes it with a delete call. Afterwards, the new story is added with add(story). This ensures that there will be no duplicate titles. All operations are wrapped in a try–catch block, and an exception is thrown in case of an error. The code therefore uses key–value operations and list searching to find and delete the old entry, then saves the new Story object into the Hive storage.

Security and offline opreation

The **AI Storybook** project pays special attention to user data security and privacy. Since all data and the generating model run locally, there is no need for an internet connection or API keys. This means that the topics provided by the user and the generated stories never leave the device. The Hive local database ensures that stories are stored securely, and because the technology is offline, all saved data is accessible only on the user's device. The fully offline mode provides strong data protection and stable performance, even in environments with weak or no network connectivity.

Limitations and challenges

Although the **AI Storybook** is an interesting experiment in creating an offline, child-friendly story generator, it has several limitations:

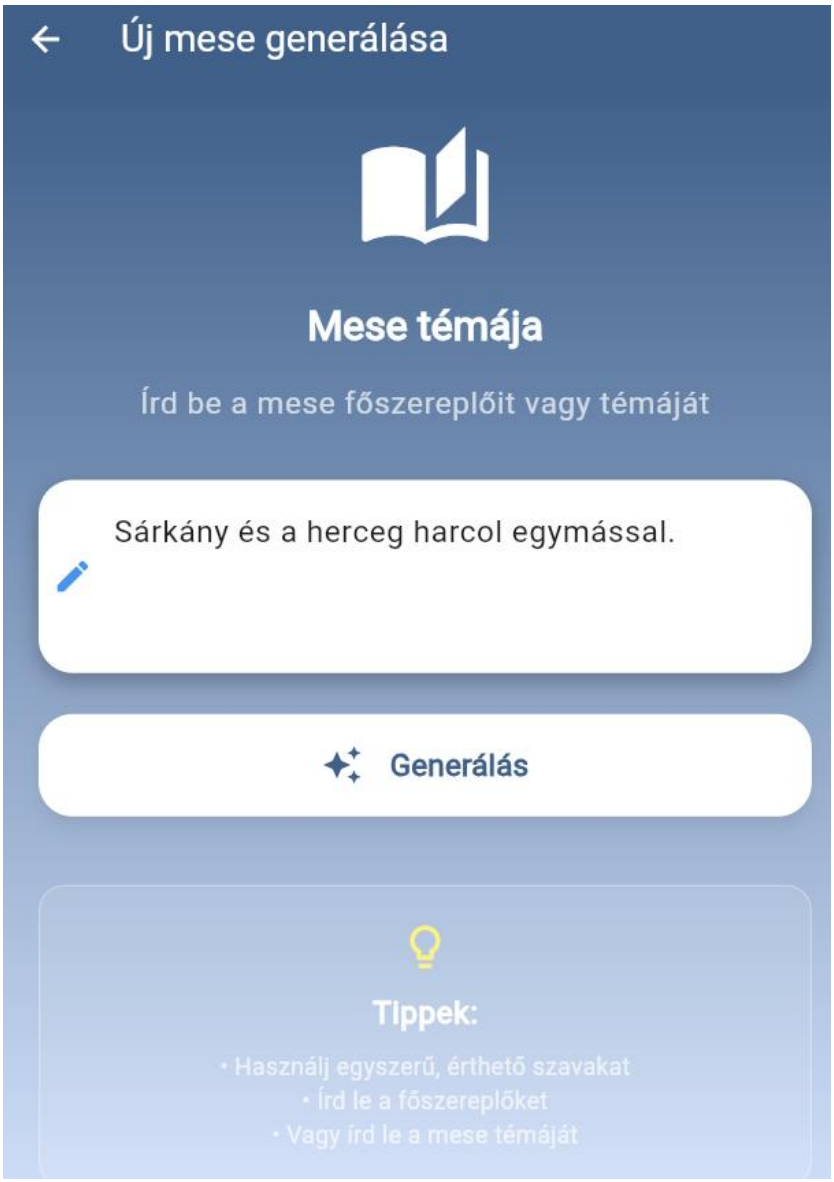
- **AI model quality:** The language models run with the Ollama framework (such as Llama2) are not specifically optimized for writing children's stories. The narratives are often somewhat fragmented, less coherent, and lack the classic storytelling structure (introduction–plot–conclusion). As a result, the generated stories are more like scattered ideas than polished, child-friendly storybook content.
- **Lack of illustrations:** The application currently does not include an image generator. The stories only come with colorful placeholder images that do not match the actual content. This is partly due to technical reasons (running a local image generator would be too resource-intensive on mobile devices) and partly for safety/child-appropriate concerns (the output of an image generator is harder to control).

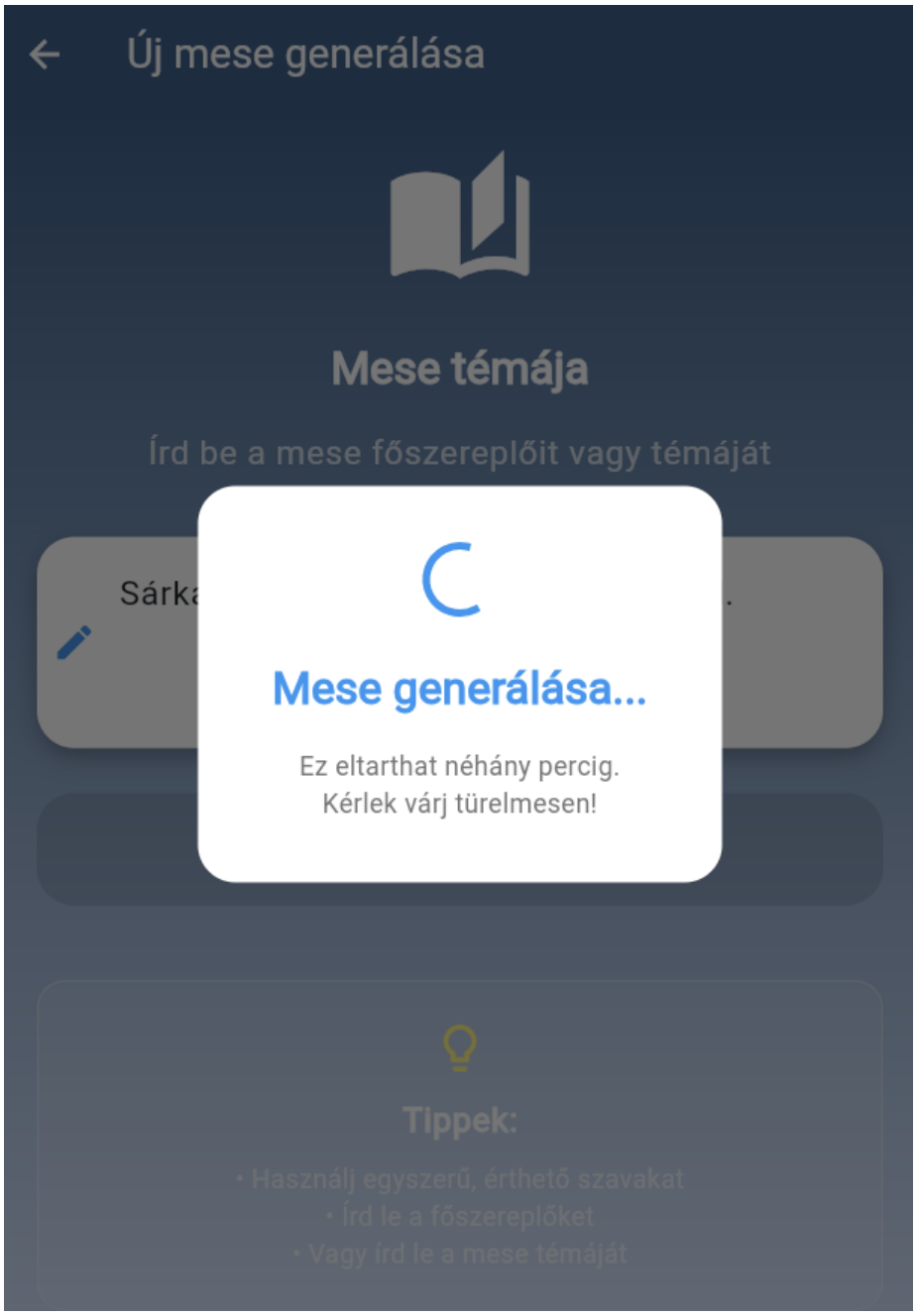
- **Simplicity vs. usability:** While the project serves well as a demonstration of how Flutter + Hive + Ollama technologies can be integrated, it is not yet suitable for real educational or entertainment purposes. The quality of the stories and the visual experience currently do not meet the expectations for content aimed at children.

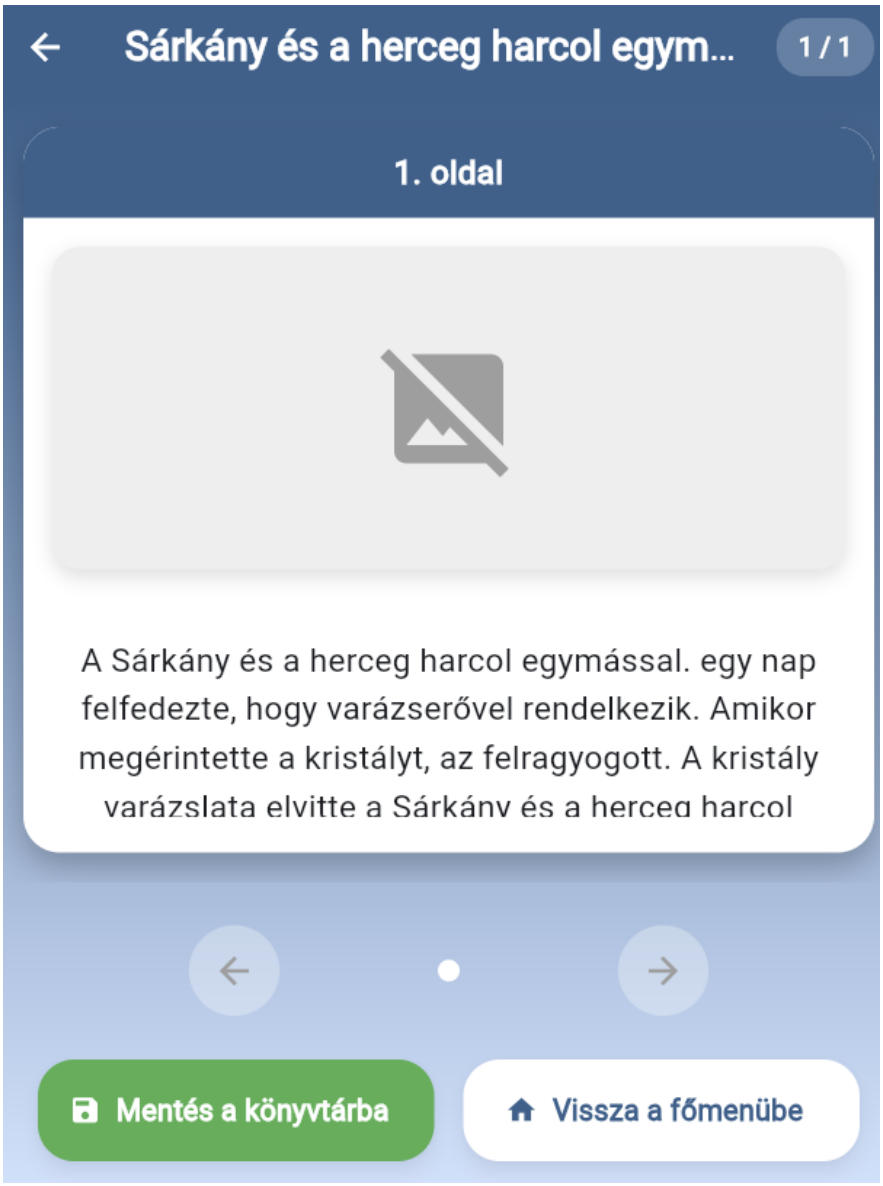
Summary

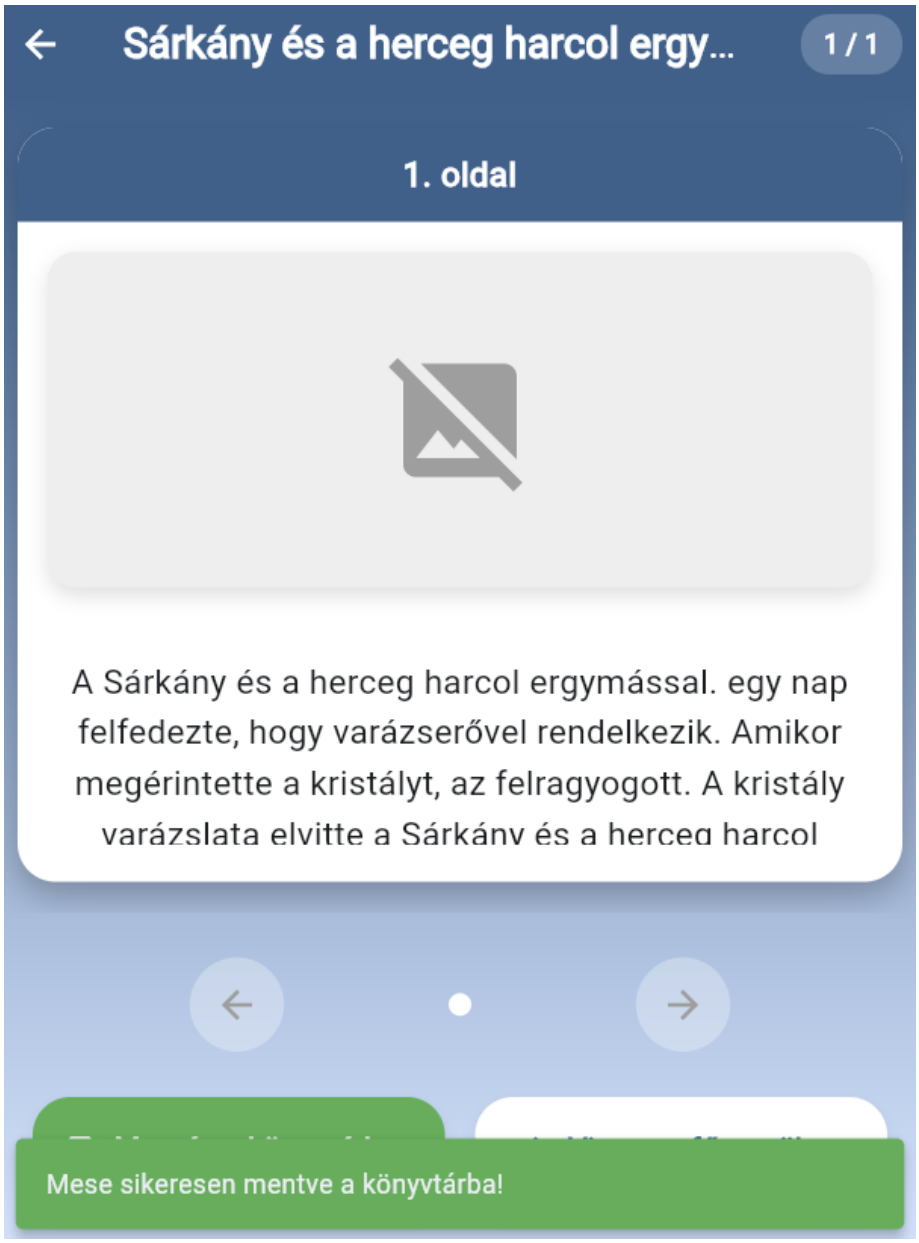
The **AI Storybook** is a child-friendly story generator application that combines the Flutter framework with modern offline AI technology (Ollama LLM). The project demonstrates how large language models can be used locally, without an internet connection, to create and manage stories. With Flutter's **PageView**, it provides an interactive storybook experience, while the **Hive** database ensures reliable local storage of the stories. The interface uses the latest **Material 3 design**, with every element tailored for easy use by children. As a showcase project, it effectively illustrates the advantages of offline AI integration and local data handling within a modern Flutter application.

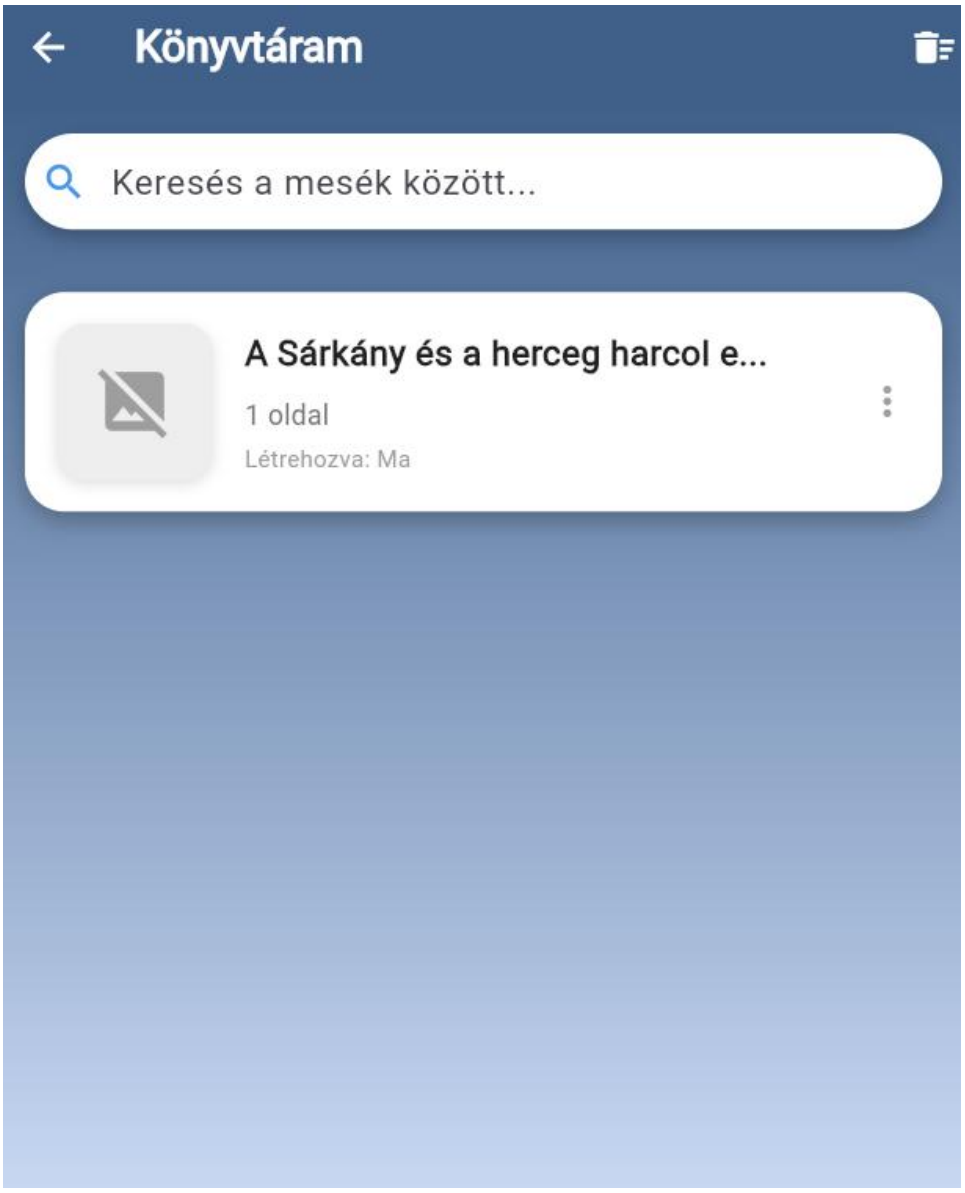
Operation


Új mese generálása
Mese témája
Írd be a mese főszereplőit vagy témáját
Sárkány és a herceg harcol egymással.
Generálás
Tippek:
• Használj egyszerű, érthető szavakat
• Írd le a főszereplőket
• Vagy írd le a mese témáját


Új mese generálása
Mese témája
Írd be a mese főszereplőit vagy témáját
Sárka
Mese generálása...
Ez eltarthat néhány percig.
Kérlek várj türelmesen!
Tippek:
• Használj egyszerű, érthető szavakat
• Írd le a főszereplőket
• Vagy írd le a mese témáját


Sárkány és a herceg harcol egym...
1 / 1
1. oldal
A Sárkány és a herceg harcol egymással. egy nap felfedezte, hogy varázserővel rendelkezik. Amikor megérintette a kristályt, az felragyogott. A kristály varázslata elvitte a Sárkány és a herceg harcol
Mentés a könyvtárba
Vissza a főmenübe


Sárkány és a herceg harcol ergy...
1 / 1
1. oldal
A Sárkány és a herceg harcol ergymással. egy nap felfedezte, hogy varázserővel rendelkezik. Amikor megérintette a kristályt, az felragyogott. A kristály varázslata elvitte a Sárkány és a herceg harcol
Mese sikeresen mentve a könyvtárba!



Created by: Zoltán Papp