LoveStorying

Team members: Maria Zhou, Brenda Sugey Vega Contreras, Pankti Shah

Version 1.1

Summary of Project

This application provides a platform for the young generation (age 3 - 12) to create stories, share stories and discover interesting stories from peers, parents, teachers or professional writers. Users can create fiction, nonfiction or educational stories by inputting characters, background and genre, and choose the length of stories. The app will then generate a story with illustrations with the help of AI. The goal is to help children develop good reading habits, so reading comprehension questions will be added to evaluate if the user has read the story. Users will also be able to share their own stories, like and provide feedback for other stories, and even report inappropriate content to ensure a safe online environment.

Project Analysis

Value Proposition

In the past few years, a decline has been seen in children enjoying reading, and parents and educators find it difficult to cultivate good reading habits. Reading is an important life skill that children must learn to excel at academics and future careers. Moreover, nowadays the Internet has a lot of inappropriate novels, animated books, and comics that can influence the spreading of hate and violence. Through LoveStorying, children will enjoy the process of creating their own stories, reading and sharing them with friends, and they will build good reading habits without their parents and teachers having to worry about the content they are consuming.

Primary Purpose

The main purpose of this project is to construct a safe online space for the young generation (age 3 - 12), where they learn from reading and creating stories with the help of AI, cultivate good learning and reading habits, as well as improve their creativity.

Target Audience

Children (ages 3 - 12): They can feel the happiness of reading and creating stories, develop reading habits, boost their imagination, creativity and curiosity.

Parents: They can monitor their children's activities and create stories for their own children very easily and keep them occupied, which is a great way to help them learn and grow.

Educators: They can transform the boring educational content into vivid stories very easily, which will invoke student’s interests and make the content much easier to understand.

Professional Writers: They can create professional works for the kids more quickly and interestingly.

Success Criteria

Phase 1: 1) Monitor time of reading stories and correct percentage of questions on the application, and evaluate reading comprehensions to assess if children are reading more; 2) Download: over 5,000 downloads; 3) User satisfaction survey: In the effective feedback, over 80% users give positive feedback.

Phase 2: 1) Perform analysis of generated content to make sure no inappropriate/violent stories are generated; 2) Download: over 100,000 downloads; 3) Financial gain: over $5,000; 4) User satisfaction survey: over 90% positive feedback.

Competitor Analysis

Competitor Summary:

Wattpad Kids offers a vast library of stories but lacks content curation, posing risks for young users.
StoryJumper focuses on book creation from page to page and publishing rather than generating content using AI, besides it is not so much community-driven storytelling like LoveStorying.
Epic! provides a curated selection of books but lacks interactive storytelling features present in LoveStorying.
Episode, Tales and Chapters are applications that allow users to craft stories using a game-like interface where the decisions made by the user alter the course and outcome of the story, but lack content curation, posing risks for young users.

LoveStorying Summary:

Strengths: LoveStorying's strengths lie in its commitment to providing a curated and safe environment for young readers, setting it apart from competitors like Wattpad Kids. The platform's integration of ChatGPT 3.5 or GPT-4 enables interactive storytelling experiences, fostering user creativity and engagement through images generated based on the story. Additionally, LoveStorying boasts an intuitive interface that caters to a diverse audience of children, parents, educators, and writers, enhancing accessibility and usability, where users can conveniently share stories amongst their community and receive feedback.

Weaknesses: While LoveStorying offers unique advantages, it faces challenges in competing with established platforms like Epic! and StoryJumper, which already possess large user bases and brand recognition. The platform may also encounter difficulties in initially building a robust library of stories compared to competitors with extensive collections. Moreover, LoveStorying will need to invest in strategic marketing efforts to overcome competition and gain traction in the market, particularly against rivals with larger marketing budgets.

Monetization Model

Phase 1: Everyone can read unlimited books on the platform for free, and earn points by platform activities like creating stories, sharing, likes, comments etc. They could use these points to exchange presents.

Phase 2: 1) Sharing story viewing income with creators. If people want to view some popular books labeled as charging (System authorizes charging permission based on viewing amount and creator to confirm charging or not), they must pay for it. 2) Selling story copyright or hardcover books or relevant products based on stories.

Initial Design

UI/UX Design

The initial design for the LoveStorying platform is shown below:

- Necessary components/interactions
  - Set user profile - parent/educator/child (for content moderation based on user type / Privacy setting)
  - Create stories
  - Read stories
  - List stories (generated by user, filter by age, genre, creation time)
    - Login as a user (implies database, password encryption)
    - Save stories (implies storage)
  - Interaction with stories (answer comprehension questions, like, comment, favorite, share, and report)
  - Send or accept friend requests for community
    - Receive notifications
    - Send friend requests or invite a friend to join (implies database, APIs for communication)
- CrucialUI
  - View story page
  - List of generated stories
  - Create new story
  - Reading Comprehension/Assessment
  - Community page
- Minimum required navigational flow
  - Dashboard
    - List of stories
    - Create new story -> Provide inputs -> Read generated story
    - Notifications
    - Communities list

- Services/APIs?
  - Authentication API to manage user registration and login
  - APIs for Generative AI (Dalle and ChatGPT)
  - APIs for communication services
  - Notification service API
  - Data retrieval from DB for listing, filtering

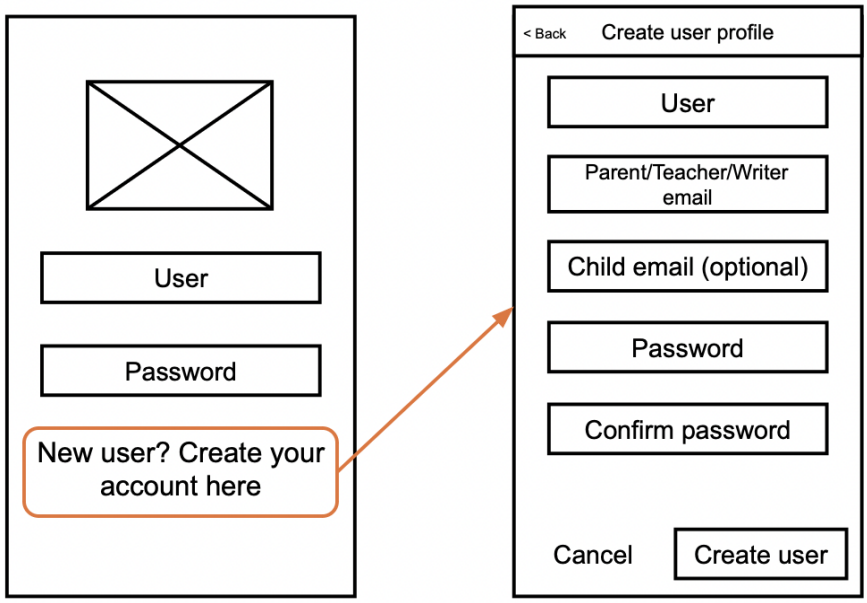

Figure 1. Sketches for login and create new user views of the LoveStorying app.

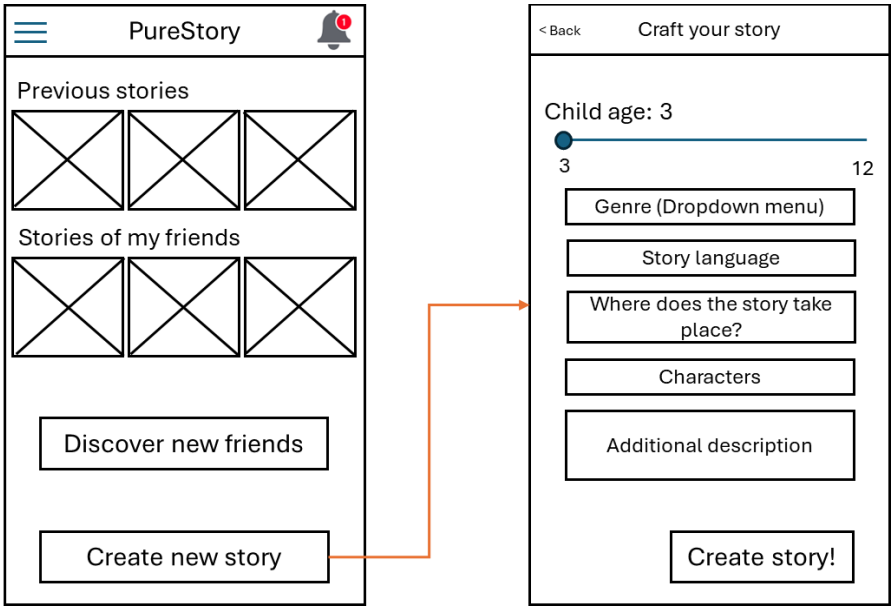

Figure 2. Sketches for Dashboard and Create new story views.

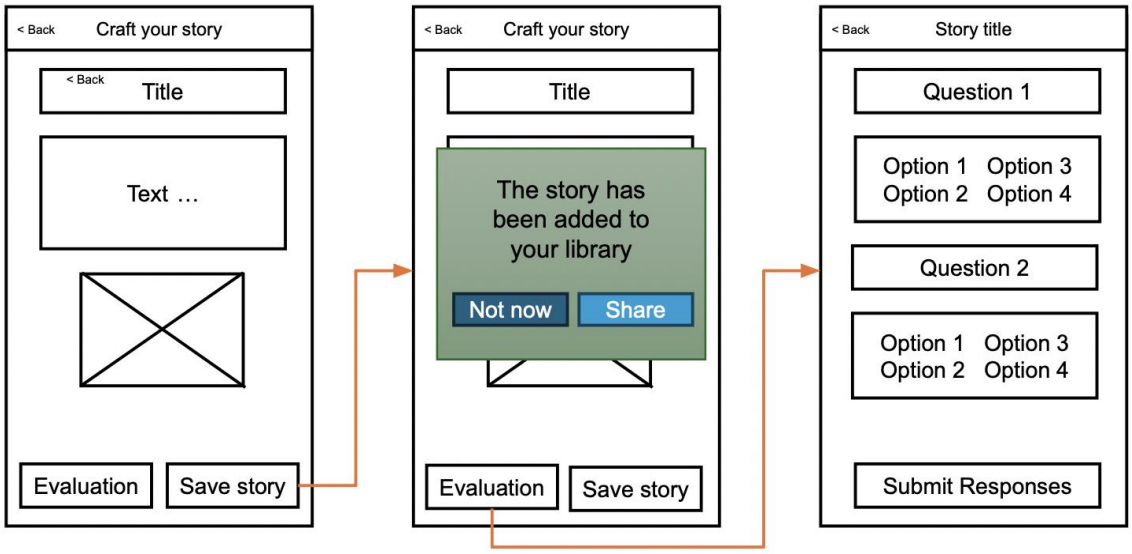

Figure 3. Sketches for view of the created story, evaluation and save option.

Technical Architecture

Three-tier architecture for the system of the storytelling app. The first tier will contain the functionality needed to access the system's services through a UI. The second tier, application tier, will be in charge of all the service requests from the client tier and accessing and processing data from the back tier. The data/service tier will contain the database and the APIs for all the additional services like ChatGPT, Dalle3, notification and communication services.

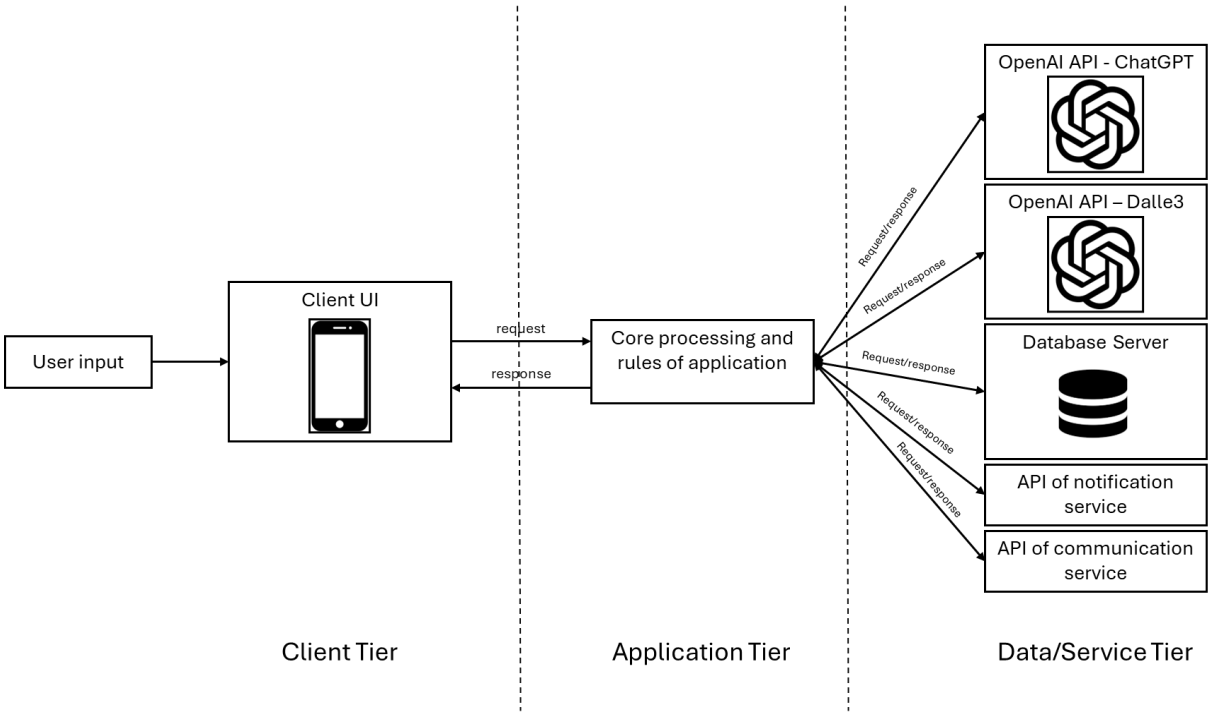

Figure 4. Three-tier architecture for the storytelling app.

Challenges and Open Questions

- **Device Compatibility** - The application should be compatible with various smartphones and tablets, which is why we intend to use a cross-platform framework for development.
- **Storage** - storing stories with pictures can be challenging as pictures will occupy more memory than plain text. Relying on free data storage will limit the amount of content we can initially store. The solution is to implement appropriate compression techniques and strategies like lazy loading or preloading content in the background for a better UX.
- **API limitations** - Free/open source APIs will have usage limits and restrictions. We will monitor API usage and implement caching strategies to reduce the number of API calls. We will explore various API options and consider using paid APIs in the next phase of development.
- **Monitoring for inappropriate content** - AI generated content needs to be moderated and it may be challenging to classify content as inappropriate. We will implement content monitoring strategies, post clear community guidelines, allow options to report inappropriate content, and try to incorporate parental controls.
- **Scalability issues** - As the user base grows, the app may experience scalability challenges, such as increased server load and database management issues. We will try to use scalable cloud infrastructure services like AWS, implement horizontal scaling techniques, and optimize database queries for performance.
- **Data Security and Privacy** - especially once monetization strategies are implemented, store user data with proper encryption strategies.
- **Offline functionality** - users may not always be connected to the internet, and some may even prefer reading in offline mode. Implement offline functionality by caching content locally on the device, allowing users to download stories for offline reading, and syncing data when the device reconnects to the internet.

Other features we would like to implement

We aim to complete the basic AI story generation in the first phase of development/prototype. Then we will proceed with content safety features to monitor for inappropriate content. To evaluate the success criteria, we will include reading comprehension style questions with stories to judge if the user has actually read. We would also try to use a text-to-speech API for children who are still learning how to read and for users with visual impairments. Then, if time permits, we will develop the user login features and community sharing options with features for likes, comments, reports etc.