Presentation 2

Overview of Presentation

Title page (take over from Pres1)

Table of contents

Introduction (take over from Pres1)

Requirements Analysis

- functional and non-functional requirements (take over from Pres1)
- data requirements
- Regulatory and Compliance Requirements
- related work
    - there are generative storybooks
    - we have an end

High-Level Design

- Architecture Overview
- Python Program
    - Book layout design prototype
    - (Page transitions) → Pres 3 ?
    - (Reader interaction) → Pres 3 ?
- AI Code Generation
    - (Integration with Python program) → Pres 3 ?
- AI Multimedia Content
    - Image, sounds generation prototype
        - sound for page transitions
        - background music (intense, balbla)
        - qr codes for more knowledge und so hintergrundwissen und so
    - (Integration with the book) → Pres 3 ?
- AI Story Generation
    - Story decision Tree
    - Core narrative creation
    - (Integration with user choices) → Pres 3 ?

Final Product (take over from Pres1)

Future Development

- what we are going to work next on
    - (Page transitions) → Pres 3
    - (Reader interaction) → Pres 3
    - (Integration with the book) → Pres 3
    - (Integration with user choices) → Pres 3
    - and prototype 2 for the rest

Focus on:

Requirement Analysis:

Functional Requirements:

- Specify the functionalities the interactive children's book should have. This could include features like sound effects, user choices affecting the story, QR code integration, and more.

Non-Functional Requirements:

- Address non-functional aspects such as performance, security, scalability, and usability. For example, the system should be user-friendly and have fast response times.

Data Requirements:

- Specify the data the system needs to function, such as the content of the books, multimedia elements, and user choices.

Regulatory and Compliance Requirements:

- If applicable, mention any legal or industry-specific regulations that need to be adhered to

High-Level Design:

Architecture Overview:

- Provide an overview of the system's architecture, explaining how different components will interact to achieve the project's goals.

Components and Modules:

- Break down the system into its major components or modules. In your case, you'd have components for the Python program, AI code generation, multimedia generation, story generation, and documentation.

Data Flow:

- Show how data will flow between components and modules. For example, how AI-generated content will be integrated into the children's book.

Interfaces:

- Describe the interfaces or APIs between components. For instance, how the Python program will communicate with AI code generation modules.

Technology Stack:

- Specify the technologies and tools that will be used for each component. This could include programming languages, AI frameworks, and multimedia libraries.

Security Considerations:

- Highlight the security measures to be implemented, especially if the project involves data collection or user accounts.

Scalability and Performance:

- Discuss how the system can scale to accommodate more users or larger books while maintaining performance.

User Experience (UX) Design:

- Outline how the user-friendly interface and interactive elements will be designed to engage young readers effectively.

Development Phases:

- Provide a rough timeline or phases for the development process, including milestones for each component.

Dependencies:

- Identify any external dependencies, such as third-party libraries or data sources, that the project relies on.