Assignment 2

Shaikhutdinov Murat

Team Members and Roles

Solo project by Shaikhutdinov Murat.
AI Tools Used: Grok for creating the LaTeX file from a text file and for grammar checking.

Use Case Diagram

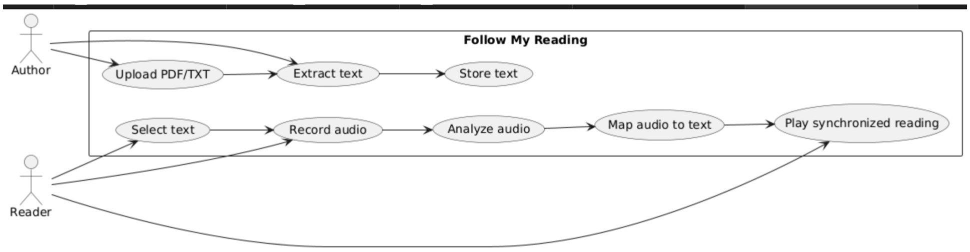

Figure 1: Use Case Diagram for the Follow My Reading Project

User Stories and Deep Product Backlog

User Stories

Must Have

- As an author, I want to upload PDF files so that I can share texts with others.
- As an author, I want to upload plain text files so that I can share content without PDF formatting.
- As a reader, I want to browse available texts so that I can choose one to practice with.
- As a reader, I want to record audio of me reading a text so that I can track my progress.
- As a user, I want to see text highlighted as audio plays so that I can follow along easily.
- As a user, I want to see different recordings of the same text so that I can compare readings.

Could Have

- As an author, I want to enable grammar checking when uploading so that my text gets automatically proofread.
- As a reader, I want to see statistics about my recording accuracy so that I can improve my reading.
- As a user, I want to bookmark texts so that I can easily return to them later.

Won't Have

- As a user, I want real-time collaborative reading sessions.
- As a user, I want AI-generated pronunciation feedback.

Backlog

Must Have (Priority 1)

1. **PDF Upload Functionality**
   - *Type*: Feature
   - *Size*: M
   - *Details*: Accept PDF files from users, extract text content using PyPDF2, store original file and extracted text.
   - *Acceptance Criteria*: Preserves basic formatting (paragraphs, line breaks).
2. **Text Upload Functionality**
   - *Type*: Feature
   - *Size*: S
   - *Details*: Accept plain text files (.txt) or direct text input.
   - *Dependencies*: Shared storage system with PDF upload.
3. **Audio Recording Interface**
   - *Type*: Feature
   - *Size*: M
   - *Details*: Telegram bot UI for recording audio, upload audio to backend.
   - *Notes*: Uses Telegram's voice message API.
4. **Audio-Text Synchronization**
   - *Type*: Feature
   - *Size*: M
   - *Details*: Whisper model transcribes audio with timestamps, map audio segments to words/phrases in text.
   - *Risks*: Accuracy depends on audio quality.
5. **Playback with Highlighting**
   - *Type*: Feature
   - *Size*: XL
   - *Details*: Mini-app highlights words in sync with audio.

Should Have (Priority 2)

1. **Reading Statistics**
   - *Type*: Feature
   - *Size*: M
   - *Details*: Track WPM (words per minute), calculate accuracy vs. reference text.
2. **Text Search**
   - *Type*: Feature
   - *Size*: S
   - *Details*: Keyword search across uploaded texts, basic ranking by relevance.

Could Have (Priority 3)

1. **Bookmarking System**
   - *Type*: Feature
   - *Size*: S
   - *Details*: Allow users to save favorite texts, list bookmarks in user profile.
2. **Recording Sharing**
   - *Type*: Feature
   - *Size*: M
   - *Details*: Generate shareable links for recordings, public/private visibility controls.

Technical Debt

1. **User Authentication**
   - *Type*: Technical
   - *Size*: M
   - *Details*: Secure Firebase auth for user accounts, Telegram login integration.
2. **Audio Processing Backend**
   - *Type*: Technical
   - *Size*: M
   - *Details*: Queue system for long-running audio jobs, retry logic for failed processing, easier to implement compared to frontend.

Emergent Items (Lower Priority)

1. **Playback Speed Control**
   - *Type*: Feature
   - *Size*: S
   - *Notes*: Requires audio pitch correction.
2. **Collaborative Reading**
   - *Type*: Feature
   - *Size*: XL
   - *Won't Have*: Marked for future phase.

Backlog Characteristics

- *Detailed Appropriately*: Near-term items (e.g., PDF upload) have clear acceptance criteria, future items (e.g., collaborative reading) are vague.
- *Estimated*: S/M/L/XL based on complexity.
- *Emerged*: Nothing prevents it
- *Prioritized*: MoSCoW method (Must/Should/Could/Won't).

Data Model

Mock API in Postman

https://www.postman.com/arin-7818637/workspace/fmr

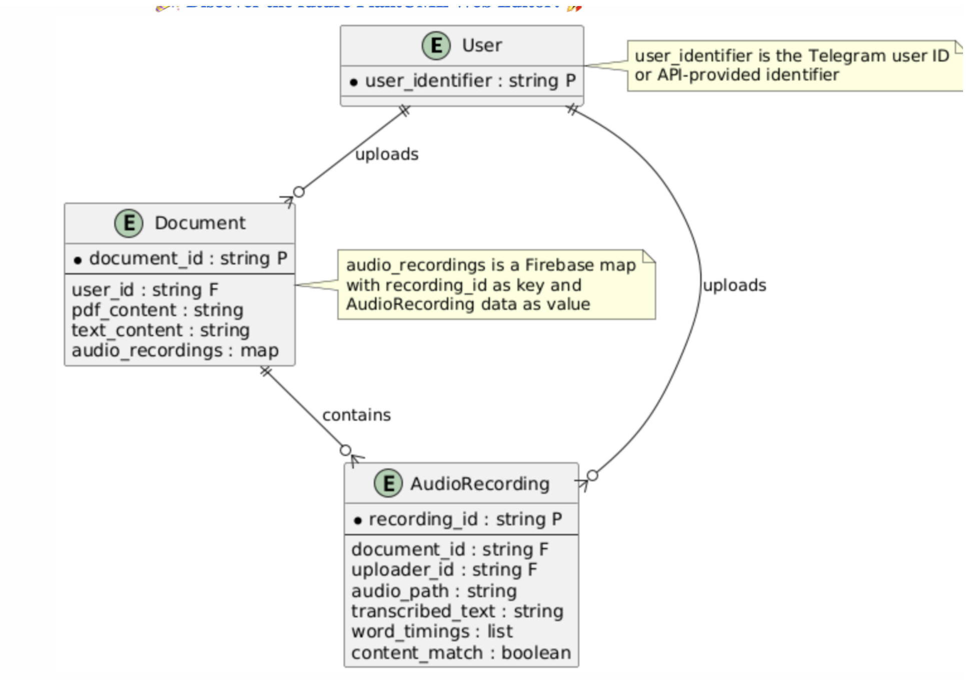

Figure 2: Entity UML Diagram for the Follow My Reading Project

Activity Tracking Sheet

Not applicable for solo project that was done before the assignments.

Summary of Key Learning

Creating the backlog helped understanding importance of prioritizing user needs using the MoSCoW method, which ensured that critical features would be implemented. This process refined my approach to product planning by focusing on user value and iterative development.