

MULTIMEDIA UNIVERSITY OF KENYA
FACULTY OF COMPUTING & INFORMATION TECHNOLOGY

PROJECT DOCUMENTATION

DEVELOPMENT OF PDF ANNOTATION AND FLASHCARD GENERATION SOFTWARE

BY
KABIRU PAUL ADAMS
CIT-227-022/2018

TR. KARIS
APRIL, 2024

Submitted in partial fulfilment of the requirements of Bachelor of Science in Software Engineering.

Declaration

I hereby declare that this Project [Proposal] is my own work and has, to the best of my knowledge, not been submitted to any other institution of higher learning.

Student : _____ Registration Number: _____

Signature: Date:.....

This project has been submitted as a partial fulfilment of requirements for the Bachelor of Science in Computer Science/Information Technology of Multimedia University of Kenya with my approval as the University supervisor.

Supervisor: _____

Signature: Date:

Abstract

Improving Learning Efficiency through Flashcard Generation from PDFs.

University students face challenges in efficiently managing and learning from vast amounts of PDF study materials. Extracting key information and creating a structured study plan can be time-consuming, leading to last-minute cramming and difficulty retaining knowledge. This research project proposes the development of a "Flashcard Generation Software" to address these inefficiencies. The software will allow users to generate flashcards directly from their PDFs, eliminating the need for manual note-taking and improving information extraction. Additionally, it will integrate a scientifically proven spaced repetition algorithm to optimise flashcard review schedules and maximise knowledge retention. By evaluating user experience and learning outcomes, this study aims to demonstrate the effectiveness of the Flashcard Generation Software in promoting a more efficient and effective learning process for students using PDF study materials.

Table of Contents

Declaration.....	2
Abstract.....	4
Table of Contents.....	5
01: Introduction.....	7
1.1 Background and context of the project.....	7
1.2 Problem Statement.....	7
1.3 Objectives and goals of the project.....	7
1.3.1 Project Objectives:.....	7
1.3.2 System Objectives:.....	8
1.4 Significance/Justification.....	8
1.5 Scope and Limitations.....	9
1.5.1 Scope of the Project:.....	9
1.5.2 Limitations of the Project:.....	9
1.6 Assumptions.....	9
02: Review of Related Work.....	10
2.1: Introduction.....	10
2.2 Spaced Repetition Research.....	10
2.3 Flashcard Design Research.....	11
2.3.1 Effective Flashcard Design Principles:.....	11
2.3.2 Incorporating Design Principles.....	11
2.4 Review of Existing Software.....	12
2.4.1 Limitations of Existing PDF Management Software for Flashcard-Based Learning.....	12
2.4.2 Proposed Solutions.....	12
03: Methodology.....	13
3.1 Introduction.....	13
3.2 Methodology for Literature Review.....	13
3.2.1 Research Objectives.....	13
3.2.2 Search Strategies.....	13
3.2.3 Selection Criteria.....	14
3.3 Methodology for Requirement Specification, Data Collection, and Analysis.....	14
3.3.1 Data Collection Techniques.....	14
3.4 Methodology for System Analysis.....	15
3.4.1 Data Flow Diagrams (DFDs).....	15
3.5 Methodology for System Design.....	16
3.5.1 User-Centred Design Principles.....	16
3.5.2 Design Process Breakdown.....	16
3.6 Methodology for System Implementation.....	17
3.6.1 Back-End Technologies.....	17
3.6.2 Front-End Technologies.....	17

3.6.3 Database Technology.....	17
3.7 Methodology for System Testing.....	17
3.7.1 Focused Testing Plan.....	17
Chapter 4: System Analysis of the Flashcard Generation Process.....	18
4.1 Description of the Current System.....	18
4.2 Feasibility Study & Conclusion.....	18
4.3 Data I/O Analysis.....	19
4.4 Process Logic Design of the Current System.....	19
4.5 Context Diagram.....	19
4.6 Chapter Summary.....	19
05: System Design.....	19
5.1 Introduction.....	19
5.2 System Description.....	20
5.3 Requirement Analysis.....	20
5.4 Conceptual Architecture.....	21
5.5 Process Logic Design.....	21
5.6 Database Design.....	23
5.8 Chapter Summary.....	23
06: Implementations and Testing.....	24
6.1 System Screenshots.....	24
6.2 Testing Plan.....	26
6.3 Evaluation Plan.....	26
6.4 Chapter Summary.....	26
07: Conclusions, Findings, and Recommendations.....	26
7.1 Introduction.....	27
7.2 Conclusions.....	27
7.3 Challenges Encountered.....	27
7.4 Future Recommendations.....	27
7.5 Conclusion.....	28
References.....	28

01: Introduction

1.1 Background and context of the project

University students face a constant struggle to efficiently manage and learn from vast amounts of study materials. Traditional methods, like relying solely on PDFs, often lead to inefficiencies. Extracting key information and creating a structured study plan from PDFs can be time-consuming, leading to last-minute cramming and difficulty retaining knowledge.

This project introduces the "Flashcard Generation Software," a solution that tackles these inefficiencies by empowering students to effortlessly generate flashcards directly from their PDFs. This eliminates the time-consuming process of manual note-taking and information extraction, allowing them to focus on understanding and retaining key concepts.

The software leverages a scientifically proven spaced repetition algorithm to schedule flashcard reviews at optimal intervals, maximising knowledge retention and promoting long-term memory.

By combining these functionalities, the "Flashcard Generation Software" aims to ease how students interact with PDF study materials, fostering a more efficient and effective learning experience.

1.2 Problem Statement

Current methods of studying with PDFs are often inefficient. Extracting key information and creating a structured study plan can be time-consuming, leading to last-minute cramming and inadequate knowledge retention. This can negatively impact exam performance and student well-being, causing stress and anxiety during the learning process.

1.3 Objectives and goals of the project

1.3.1 Project Objectives:

- **Enhance Learning Efficiency through Flashcard Generation:** Develop software that simplifies the process of creating flashcards from PDFs, enabling users to focus on learning instead of manual note-taking.
- **Improve Knowledge Retention with Spaced Repetition:** Integrate a scientifically proven spaced repetition algorithm to schedule flashcard reviews and maximise long-term memorization of key concepts.
- **Promote Accessibility with Cross-Platform Compatibility:** Ensure the software functions seamlessly across various devices and operating systems (e.g., Windows, macOS, Android, iOS) to cater to diverse user preferences.

- **Prioritise User Experience with an Intuitive Interface:** Design a user-friendly and customizable interface that fosters a smooth learning experience and simplifies interaction with the software's functionalities.

1.3.2 System Objectives:

- **Effortless PDF Integration and Management:** The software should seamlessly import and organise PDFs, allowing users to easily access and interact with their study materials.
- **Enhanced Information Extraction and Flashcard Creation:** Implement features that facilitate efficient identification and extraction of key information from PDFs, streamlining the process of flashcard creation.
- **Prioritise User Privacy and Data Security:** Integrate robust security measures to ensure the confidentiality and protection of user data, including study materials and flashcard content.
- **Learning Progress Insights and Analytics:** Provide users with data-driven insights into their learning progress, such as flashcard review statistics and identification of knowledge gaps.
- **Comprehensive User Support System:** Establish a user support system encompassing readily available tutorials, FAQs, and a clear feedback mechanism for users to report issues and suggest improvements.

1.4 Significance/Justification

Traditional methods of studying with PDFs often lead to inefficiencies. Extracting key information and creating a structured study plan can be time-consuming, leading to last-minute cramming and difficulty retaining knowledge. This can negatively impact student performance and well-being, causing stress and anxiety.

The "Flashcard Generation Software" addresses these challenges by empowering students to generate flashcards directly from their PDFs. This eliminates the need for manual note-taking and allows them to focus on understanding and retaining key concepts. The software's integrated spaced repetition algorithm ensures optimal flashcard review schedules, maximising long-term knowledge retention.

By combining efficiency with effective learning methods, this software has the potential to revolutionise the way students interact with PDFs. It can promote a more organised and efficient study process, leading to improved academic performance, reduced stress, and increased confidence during exam preparation.

1.5 Scope and Limitations

1.5.1 Scope of the Project:

- **Seamless PDF Integration:** The software will allow users to import, organise, and manage their PDF study materials for easy access and navigation.
- **Intelligent Information Extraction:** Users can leverage built-in tools to identify and extract key information from PDFs, facilitating efficient flashcard creation.
- **Flashcard Generation with Spaced Repetition:** The software will integrate a scientifically proven spaced repetition algorithm to automatically schedule flashcard reviews and optimise knowledge retention.

1.5.2 Limitations of the Project:

- **Phased Feature Implementation:** Due to project scope and time constraints, not all potential features may be implemented initially. Prioritisation will be based on user needs and impact on learning effectiveness.
- **Limited External Integrations:** The initial development phase will focus on core functionalities like PDF management, flashcard generation, and spaced repetition. Integrations with third-party apps or services may be considered in future updates.
- **Resource Dependence:** Project development is subject to the availability of resources such as funding and development time. This may influence the pace or scope of feature implementation.

1.6 Assumptions

- **Technical Literacy:** We assume users have a basic understanding of using software applications on their chosen devices (computer, phone, etc.).
- **PDF Compatibility:** We assume the software can effectively process a wide range of standard PDF formats. Highly complex or scanned PDFs with intricate layouts may require additional development efforts.
- **Learning Styles:** While flashcards are a valuable learning tool, we acknowledge that some students may benefit from additional study methods tailored to their individual learning styles.
- **User Motivation:** The software is designed to facilitate efficient studying, but we assume users will put in the necessary effort to utilise the flashcards and spaced repetition features effectively.

02: Review of Related Work

2.1: Introduction

The vast amount of information available today, often presented in PDFs, creates a challenge for students. Extracting key points and effectively learning from these documents can be time-consuming and hinder knowledge retention. While valuable for general knowledge management, existing personal knowledge management (PKM) systems often require significant customisation and lack features specifically tailored for efficient learning from PDFs. Students struggle with tasks like manual note-taking and difficulty highlighting key information for later review.

This literature review focuses on exploring solutions to address these challenges. By delving into existing knowledge management systems and their limitations in the context of studying from PDFs, we aim to extract insights that will inform the development of a "Flashcard Generation Software." This software aims to streamline the learning process from PDFs by automatically extracting key information and facilitating knowledge retention through the use of flashcards.

This section explores existing literature and software related to:

- **Spaced Repetition:** We will examine research on spaced repetition algorithms and their effectiveness in enhancing long-term knowledge retention.
- **Flashcard Design:** We will analyse research on effective flashcard design principles, focusing on factors that can influence learning outcomes.
- **PDF Management and Annotation Tools:** We will review existing software solutions for managing and annotating PDFs, identifying their limitations in facilitating efficient learning.

2.2 Spaced Repetition Research

Spaced repetition algorithms address the limitations of cramming by strategically scheduling information reviews at optimal intervals based on forgetting curves. Compared to cramming methods, this approach strengthens memory consolidation and promotes long-term knowledge retention.

One promising spaced repetition algorithm is the Free Spaced Repetition Scheduler (FSRS). FSRS utilises a variant of the Difficulty, Stability, Retrievability (DSR) model to personalise review schedules for each flashcard based on user performance (FSRS4Anki wiki). This personalisation ensures optimal learning efficiency by focusing on concepts the user finds challenging.

By integrating a scientifically proven spaced repetition algorithm, our Flashcard Generation Software will optimise flashcard review schedules to maximise user learning outcomes and knowledge retention.

2.3 Flashcard Design Research

This section explores research on effective flashcard design principles to enhance learning outcomes for users of our Flashcard Generation Software. By understanding these principles, the software can be designed to generate flashcards that optimise knowledge retention and learning from PDFs.

2.3.1 Effective Flashcard Design Principles:

Research suggests that flashcards with lower information density, clear organisation, minimal extraneous details, and high visual search ability lead to better knowledge retention (Pyc & Rawson, 2011). Additionally, a clear and concise presentation involving well-organised content, minimal text, and appropriate font size and spacing promotes learning (Pyc & Rawson, 2011). While the ideal font type for flashcards might require further investigation, research by Gasser et al. (2005) highlights the importance of user-friendliness and avoiding overly ornate or difficult-to-read fonts.

2.3.2 Incorporating Design Principles

Our Flashcard Generation Software will be designed to consider these research findings on effective flashcard design. The software will allow users to:

- **Customise content structure:** Users will have the option to adjust font size, spacing, and information density on flashcards to optimise clarity and conciseness. The software could potentially analyse the extracted information from PDFs and suggest optimal content structure based on information density principles.
- **Select question types:** The software will offer various question formats, such as multiple choice, true/false, and open-ended, allowing users to tailor the format to their learning objectives.
- **Highlight key information:** The software could employ text summarization techniques or allow users to highlight key points within the PDF to focus on essential information for flashcard creation.
- **Integrate visuals:** Users will have the ability to incorporate visuals like diagrams and images extracted from PDFs or uploaded directly to further enhance learning and memory retention.

By incorporating these design features, our Flashcard Generation Software will empower users to create effective and engaging flashcards that optimise their learning experience from PDFs.

2.4 Review of Existing Software

This section assesses current software solutions for PDF management and annotation, focusing on their suitability for efficient learning. While popular tools like Adobe Acrobat and Foxit Reader offer valuable features, they often fall short in facilitating streamlined information extraction and flashcard creation for optimal learning.

2.4.1 Limitations of Existing PDF Management Software for Flashcard-Based Learning

A variety of software solutions exist for managing and annotating PDFs, such as Adobe Acrobat and Foxit Reader. While these tools offer functionalities like highlighting, note-taking, and searching, they often lack features specifically designed for efficient learning from PDFs. Key limitations include:

- **Manual Information Extraction:** These tools typically require users to manually highlight or copy key information from the PDF, a time-consuming process that can hinder learning efficiency.
- **Limited Flashcard Creation Features:** While some tools allow adding notes or annotations, they lack functionalities to easily convert highlighted text or notes into well-structured flashcards.

2.4.2 Proposed Solutions

These limitations underscore the necessity for software explicitly designed to aid learning from PDFs. Our proposed Flashcard Generation Software aims to bridge this gap by offering:

- **Automated Information Extraction:** Utilising text summarization techniques or user-defined parameters, the software will automatically extract key information from PDFs, saving users time and effort.
- **Streamlined Flashcard Creation:** Users will be able to convert extracted information or highlighted text into well-structured flashcards with customizable question formats and visual integration.
- **Spaced Repetition Integration:** The software will incorporate spaced repetition algorithms to schedule flashcard reviews at optimal intervals, thereby maximising knowledge retention and learning outcomes.

03: Methodology.

3.1 Introduction

This chapter outlines the methodology employed for the individual execution of the project to meet the defined objectives. The chosen approach is tailored to ensure a systematic and effective progression of the project, considering the solitary nature of its execution.

3.2 Methodology for Literature Review

This section outlines the research methods employed to gather and analyse relevant information for the design and development of the Flashcard Generation Software.

3.2.1 Research Objectives

The literature review aims to achieve the following objectives:

- Understand effective flashcard design principles related to information density, question types, and visual elements. This will guide the design of functionalities within the software to promote optimal knowledge retention.
- Explore research on limitations of existing PDF management software for flashcard-based learning. Identifying these limitations will highlight the need for the Flashcard Generation Software and its unique functionalities.
- Analyse the benefits and applications of spaced repetition algorithms for knowledge retention. Integrating a spaced repetition algorithm into the software will optimise flashcard review schedules and learning outcomes.

3.2.2 Search Strategies

To locate relevant research papers, the following resources and methods will be employed:

- **Academic Databases:** A search will be conducted in the following academic databases:
 - ScienceDirect: This database offers a vast collection of scholarly articles in Science, Technology, Medicine, and Social Sciences. Keyword searches will be conducted using terms like "flashcard design," "information density," "spaced repetition," and "PDF management software."
 - ERIC (Education Resources Information Center): This database focuses on educational research and resources. Searches will target terms like "flashcard-based learning," "educational software," and "PDF learning tools."

- **Search Engines:** Google Scholar will be used to broaden the search scope and identify additional relevant publications, including conference proceedings and grey literature (unpublished research reports). The keywords used will be similar to those used in academic databases.

3.2.3 Selection Criteria

The following criteria will be used to evaluate and select relevant research papers:

- **Publication Date:** Articles published within the past 5-10 years will be prioritised to ensure access to the most up-to-date research findings.
- **Peer-Reviewed Articles:** Priority will be given to peer-reviewed articles from academic journals, as they undergo a rigorous review process for quality and methodological soundness.
- **Relevance to Research Objectives:** The research must directly address one or more of the defined research objectives and provide valuable insights for the design and development of the Flashcard Generation Software.

3.3 Methodology for Requirement Specification, Data Collection, and Analysis

This section outlines the methods used to gather and analyse user requirements for the Flashcard Generation Software. By understanding user needs and expectations, the software can be designed to effectively address their learning goals and provide a valuable learning experience.

3.3.1 Data Collection Techniques

To gather comprehensive user requirements for the Flashcard Generation Software, a combination of the following data collection techniques will be employed:

- **User Interviews:** Semi-structured interviews will be conducted with a target group of students who represent the primary users of the software. The interview guide will consist of open-ended questions designed to explore:
 - Current challenges students face with studying from PDFs (e.g., difficulty highlighting key points, inefficient note-taking)
 - Preferred methods for creating flashcards from learning materials
 - Desired functionalities for flashcard generation software (e.g., automatic information extraction, customisable question formats)
 - User expectations for the software's interface and ease of use
 Interviews can be conducted in person, online, or through video conferencing, depending on participant availability and preferences.

- **Questionnaires:** An online questionnaire will be developed and distributed to a broader student population. The questionnaire will include a mix of:
 - Multiple-choice questions to gather data on preferred question formats (e.g., multiple choice, true/false, open-ended) and desired functionalities.
 - Likert scale questions to gauge user preferences for features like information density control and visual integration options.
 - Open-ended questions allow users to express additional thoughts, suggestions, or challenges related to using flashcards for studying from PDFs.

By combining user interviews and questionnaires, the project will gather both in-depth qualitative data and broader quantitative data to gain a comprehensive understanding of user needs and preferences.

3.4 Methodology for System Analysis

This section outlines the methodologies used to analyse the current system, which likely involves manual processes for creating flashcards from PDFs. Understanding the limitations and inefficiencies of the current system will highlight the need for the Flashcard Generation Software.

3.4.1 Data Flow Diagrams (DFDs)

Data Flow Diagrams (DFDs) will be a primary tool used to visually represent the flow of information within the current system (manual flashcard creation from PDFs). Here's a breakdown of how DFDs will be utilised:

- **Context Diagram:** A high-level context diagram will be created to illustrate the current system's interaction with external entities. In this case, the external entities would be the user and the PDF document. The diagram will show the user as the source of information (needing to create flashcards) and the PDF as the data source (containing the information for flashcards).
- **Level 0 DFD:** A Level 0 DFD will be developed to depict the main process involved in creating flashcards from PDFs using the current system. This process might involve:
 - User Input: The user searches for and opens a PDF document.
 - Manual Information Extraction: The user manually highlights, summarises, or rewrites key points from the PDF to create flashcard content.
 - Flashcard Creation (Manual): The user manually creates flashcards using physical notecards, a digital note-taking application, or another manual method.

3.5 Methodology for System Design

This section outlines the methodology for designing the Flashcard Generation Software. We prioritise a user-centred approach while acknowledging the practical constraints of a student project.

3.5.1 User-Centred Design Principles

The system design will adhere to the following user-centric principles:

- **Intuitive User Interface (UI):** The UI will be simple and easy to use, with a focus on essential functionalities such as uploading PDFs, selecting information, and creating flashcards.
- **Prioritizing Core Functionalities:** We will prioritize features that directly address the needs of users. These may include processing PDFs, extracting information, creating flashcards, and potentially incorporating a basic spaced repetition algorithm to schedule reviews.
- **Flexibility within Constraints:** We will explore options to allow users to define parameters, such as keywords or highlighting, for information extraction. Depending on available resources, we may investigate incorporating more advanced Natural Language Processing (NLP) techniques.

3.5.2 Design Process Breakdown

Here's a breakdown of the key design phases we will follow:

1. **User Requirements Analysis:** We will revisit and refine the user requirements identified earlier. This ensures the design aligns with user needs and project feasibility.
2. **System Architecture Definition:** We will define the software's core components and their interactions. This will include the UI, PDF Processing Module, Information Extraction Module, Flashcard Creation Module, Database, and (optional) Spaced Repetition Algorithm module.
3. **User Interface (UI) Design:** We will prioritise a user-friendly UI design that facilitates core functionalities. This might involve features for uploading PDFs, selecting relevant information, and creating and customising flashcards.
4. **Information Extraction Design:** We will plan how the software will identify key information from uploaded PDFs. This might involve user-defined parameters for basic extraction or exploring NLP techniques if project scope allows.

5. **Database Design:** We will design a database to store essential information, such as user information, extracted information from PDFs, and user-created flashcards. We will consider storage constraints when deciding whether to store uploaded PDFs themselves.

3.6 Methodology for System Implementation

This section describes the chosen approach for implementing the Flashcard Generation Software.

3.6.1 Back-End Technologies

The software's back-end functionalities will be developed using the following technologies:

- Programming Language: Typescript
- Web Framework: Next.js
- Libraries:
 - PDF processing: PSPDFKit
 - Information extraction: GPT-4
 - Database interaction: Prisma ORM

3.6.2 Front-End Technologies

The user interface will be built using:

- Front-End Frameworks: Next.js, Tailwind CSS, shadcn/ui

3.6.3 Database Technology

The system will utilise a PostgreSQL database managed by Prisma ORM.

3.7 Methodology for System Testing

While a comprehensive testing plan is ideal, this project will prioritise core functionalities due to time limitations. The following outlines a streamlined testing approach to ensure the Flashcard Generation Software meets basic usability and functionality requirements.

3.7.1 Focused Testing Plan

The testing plan will focus on critical aspects of the software:

- **Core Functionality Testing:** Verify core functionalities like PDF upload, information extraction, flashcard creation, and spaced repetition algorithm integration work as intended. Manual testing will be the primary method for this phase.

- **Basic Usability Testing:** Conduct informal usability testing with a small group of users to identify any major usability issues that impede core tasks. Techniques like walkthroughs and user feedback sessions can be employed.

Chapter 4: System Analysis of the Flashcard Generation Process

This chapter analyses the existing manual process of creating flashcards from PDFs. It identifies the system's strengths and weaknesses, evaluates its feasibility for improvement, and outlines the data flow and logic involved.

4.1 Description of the Current System

This section describes the current manual process of creating flashcards from PDFs. Here's a breakdown:

- **Strengths:**
 - Users have complete control over the information selected for flashcards.
 - No dependence on external software or specific hardware requirements.
- **Weaknesses:**
 - Time-consuming and repetitive, especially for large PDFs.
 - Prone to errors during manual information extraction and transcription.
 - Limited formatting options for creating flashcards.

4.2 Feasibility Study & Conclusion

A feasibility study was conducted to assess the viability of developing a software solution to automate flashcard creation from PDFs. The study considered the following factors:

- **Technical Feasibility:** Modern programming languages and libraries offer functionalities for processing PDFs, extracting text, and building user interfaces. Developing the Flashcard Generation Software is technically feasible.
- **Economic Feasibility:** The project scope focuses on creating a student-oriented tool, and development costs are limited. The potential benefits in terms of time saved and improved study efficiency outweigh the development investment.
- **Operational Feasibility:** The software is designed to be user-friendly and requires minimal technical expertise to operate. Integrating it into existing workflows should not pose significant challenges.

Conclusion: The feasibility study indicates that developing the Flashcard Generation Software is a viable solution to address the limitations of the current manual process.

4.3 Data I/O Analysis

This section analyses the data involved in the current manual process:

- **Data Captured:** The primary data input is the PDF document containing the information for flashcards.

- **Data Relationships:** There are no explicit data relationships between captured information, as users manually select relevant sections for flashcards.
- **Outputs:** The output of the current system is a set of handwritten or digital flashcards created from the extracted information within the PDF.

4.4 Process Logic Design of the Current System

This section outlines the logic flow of the current manual process using a flowchart:

4.5 Context Diagram

The context diagram depicts the system's interaction with external entities. In this case, the external entity is the user who interacts with the manual process of creating flashcards from a PDF document.

4.6 Chapter Summary

This chapter analysed the current system of manually creating flashcards from PDFs. It identified the strengths and weaknesses of the existing process, concluded that developing a software solution is feasible, and analysed the data flow and logic involved. The information gathered in this system analysis chapter serves as the foundation for designing the Flashcard Generation Software in the following chapter.

05: System Design

This chapter outlines the design choices and architecture of the Flashcard Generation Software. It details the system's functionalities, analyses requirements, and presents the conceptual design using various modelling techniques.

5.1 Introduction

This chapter dives into the design of the Flashcard Generation Software. We've adopted a user-centred design approach, prioritising functionalities that directly address user needs for creating flashcards from PDFs while acknowledging the limitations of a student project.

5.2 System Description

The Flashcard Generation Software is designed to assist students in creating flashcards from PDFs. Users can upload PDFs, select relevant information for flashcards (either through highlighting or basic keyword selection - depending on implemented functionalities), and customise the creation of flashcards. The software aims to streamline the process of flashcard creation from PDFs, saving students time and effort compared to manual methods.

- **Strengths:** The software offers a user-friendly interface for creating flashcards, reduces manual effort and time spent on flashcard creation, and has the potential to improve study efficiency by allowing users to focus on the content rather than the creation process.
- **Weaknesses:** Due to project scope limitations, the current implementation might have limitations in information extraction capabilities (potentially relying on user input for highlighting or keywords). Advanced features like more sophisticated NLP techniques for information extraction or complex spaced repetition algorithms might be explored in future iterations.

5.3 Requirement Analysis

Here's a breakdown of the requirements for the Flashcard Generation Software:

- **Functional Requirements:**
 - The software should allow users to upload PDFs.
 - It should provide functionalities to extract relevant information for flashcards (either through user-defined parameters or basic NLP techniques, depending on implementation).
 - Users should be able to create flashcards with basic customization options (text size, font).
 - (Optional) The software might offer functionalities for exporting flashcards in different formats.
- **Non-Functional Requirements:**

- The software should be user-friendly and easy to navigate.
- It should function reliably with minimal errors within the expected usage patterns.
- Security considerations will prioritise user privacy for uploaded PDFs and stored information within the chosen project scope (e.g., secure storage practices if PDFs are stored).
- **User Requirements:**
 - The software should address the user's need to create flashcards from PDFs efficiently.
 - It should provide a user interface that is intuitive and easy to learn for students with varying technical backgrounds.
 - The software should offer basic functionalities to customise flashcards and cater to individual study styles.
- **Usability Requirements:**
 - The user interface should be clean and uncluttered, with clear instructions for using core functionalities.
 - The software should be responsive and provide feedback to users during interactions (e.g., progress indicators during PDF processing).
 - Users should be able to navigate the software easily and complete tasks like uploading PDFs and creating flashcards with minimal steps.

5.4 Conceptual Architecture

The Flashcard Generation Software is designed with the following core components:

- **User Interface (UI):** This is the primary point of interaction for users. It allows them to upload PDFs, select information for flashcards, customise flashcard options, and view generated flashcards.
- **PDF Processing Module:** This module handles tasks related to uploaded PDFs, including extracting text and performing basic formatting and cleaning (if applicable).
- **Information Extraction Module (Optional):** Depending on the implemented functionalities, this module might use user-defined parameters (keywords or highlighting) for basic information extraction. Exploration of Natural Language Processing (NLP) techniques for more advanced information identification can be considered for future development.
- **Flashcard Creation Module:** This module allows users to convert extracted information or highlighted text passages into customisable flashcards. Users can define options like question format, information density, and visual elements (if implemented).
- **Spaced Repetition Algorithm (Optional):** A basic spaced repetition algorithm could be explored for scheduling flashcard reviews, but its implementation might depend on project scope and available resources.
- **Database:** The software utilises a database to store essential information, such as:
 - User information

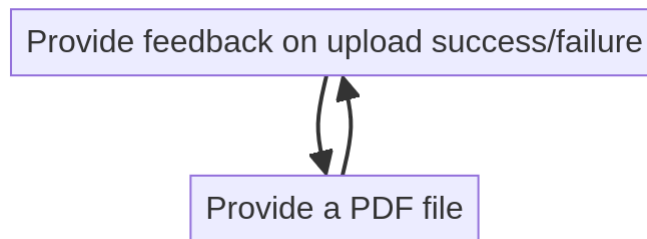
- Uploaded PDFs (optional, depending on storage constraints)
- Extracted information from PDFs
- User-created flashcards

5.5 Process Logic Design

Here, we will utilise various modelling techniques to depict the system's logic and data flow. We will focus on the most relevant techniques for this project and provide clear explanations for each:

Data Flow Diagrams (DFDs):

- **Context Level DFD:** This will illustrate the system's interaction with external entities. The user and the PDF document will be the external entities. The data flows between them could be:
 1. User -> System: Provide a PDF file
 2. System -> User: Provide feedback on upload success/failure (e.g., "PDF uploaded successfully" or "Error: Invalid file format")

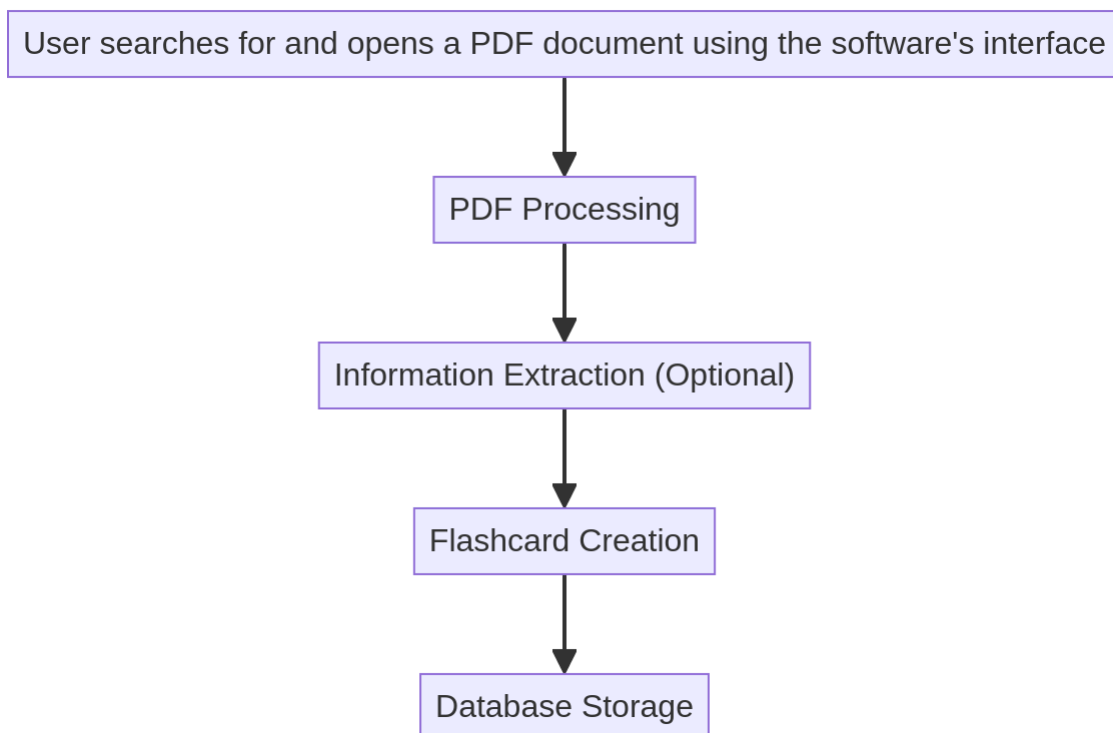


- **Level 0 DFD:** This DFD will depict the main process of creating flashcards from PDFs. Here's a breakdown of the processes involved:

Assumptions: This process assumes the user has basic computer skills and can navigate a standard file selection dialog.

1. **User Input:** The user searches for and opens a PDF document using the software's interface.
2. **PDF Processing:** The PDF Processing Module extracts text from the uploaded PDF and performs basic formatting or cleaning (if applicable).
3. **Information Extraction (Optional):** Depending on the implemented functionalities, this step might involve:

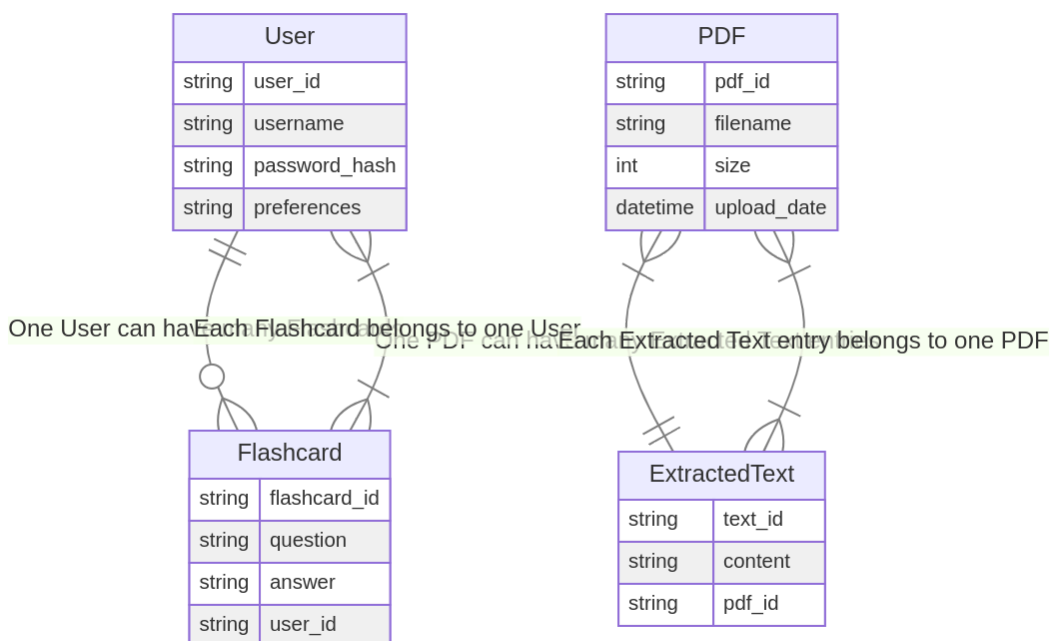
- **User-defined Selection:** The user highlights specific sections within the PDF that contain relevant information for flashcards.
 - **Keyword Selection (Optional):** The user provides keywords to guide the system in identifying relevant information from the extracted text. (This might be a more advanced option depending on the chosen NLP techniques). **For example**, the user might provide keywords like "photosynthesis," "mitochondria," and "cellular respiration" to help the system identify relevant sections from a biology textbook PDF.
4. **Flashcard Creation:** The Flashcard Creation Module utilises the extracted information or highlighted text passages to generate flashcards. Users can customise options like:
- Question format (multiple choice, true/false, open-ended)
 - Information density (amount of information presented on each card)
 - Visual elements (if implemented, allowing users to add images or diagrams)
5. **Database Storage:** The system stores essential information in the database, such as user information, extracted text from PDFs, and user-created flashcards. (Uploaded PDFs might be stored depending on project scope and storage constraints).



5.6 Database Design

The Flashcard Generation Software utilises a relational database to store and manage data. Here's an overview of the database design:

- **Entity-Relationship Diagram (ERD):** The ERD will visually represent the entities (data objects) within the system and their relationships. Here are the potential entities:
 - User: This entity will store information about registered users, including username, password (securely hashed), and any relevant user preferences. (One User can have many Flashcards - one-to-many relationship).
 - PDF (Optional): This entity might store information about uploaded PDFs, such as filename, size, and upload date (depending on the decision to store PDFs themselves). (One PDF can have many Extracted Text entries - one-to-many relationship).
 - Extracted Text: This entity will store the extracted text content from uploaded PDFs. (One Extracted Text entry belongs to one PDF - many-to-one relationship).
 - Flashcard: This entity will store information about each flashcard created by the user, including:
 - User ID (foreign key referencing the User entity)
 - Text (the content of the flashcard)
 - Question Format (e.g., multiple choice, true/false)
 - Additional fields can be added to store information on visual elements (if implemented) or scheduling data for spaced repetition (if applicable).



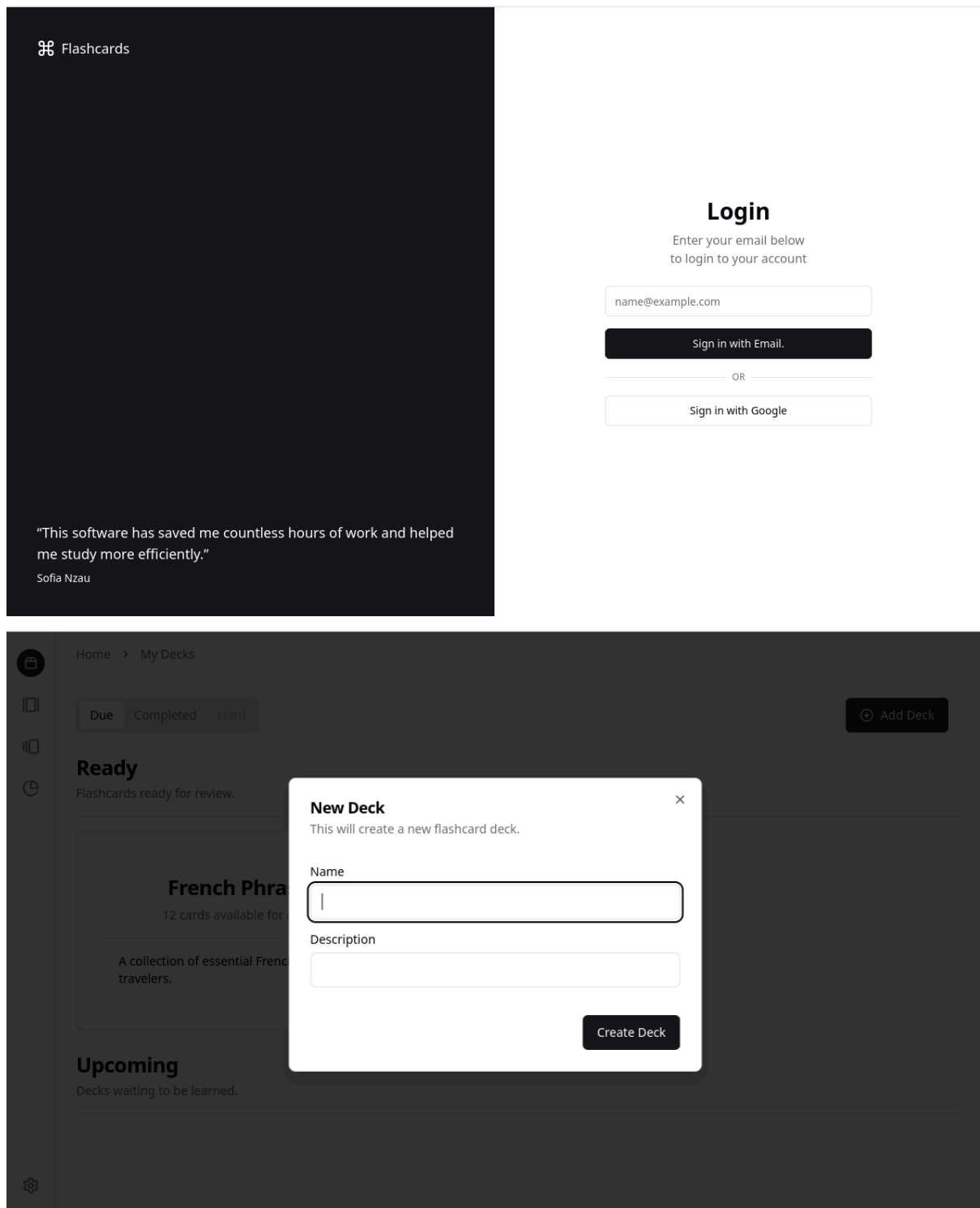
5.8 Chapter Summary

This chapter outlined the system design of the Flashcard Generation Software. It described the core functionalities, analysed various requirements, and presented the conceptual architecture using relevant modelling techniques. The chapter also provided insights into the database design and user interface considerations. The chosen design approach prioritises a user-centred experience while acknowledging the constraints of a student project. This foundation lays the groundwork for future development and exploration of advanced features like more sophisticated NLP techniques for information extraction.

06: Implementations and Testing

This chapter details the implementation process of the Flashcard Generation Software, the testing methodologies employed to ensure functionality and usability, and the evaluation plan for assessing the software's effectiveness.

6.1 System Screenshots



Home
 >
 My Decks

All Decks

Find your decks.

Status

Priority

View

<input type="checkbox"/>	Task	Status	Priority	
<input type="checkbox"/>	TASK-8782	In Progress	→ Medium	...
<input type="checkbox"/>	TASK-7878	Backlog	→ Medium	...
<input type="checkbox"/>	TASK-7839	Todo	↑ High	...
<input type="checkbox"/>	TASK-5562	Backlog	→ Medium	...
<input type="checkbox"/>	TASK-8686	Canceled	→ Medium	...
<input type="checkbox"/>	TASK-1280	Done	↑ High	...

Upload a PDF Document

Easily upload and preview your PDF documents. Drag and drop or click to select a file.

Drag and drop a PDF file or click to select

Select PDF

6.2 Testing Plan

Given the time constraints of a student project, this section will outline a streamlined testing approach focusing on core functionalities.

- **Core Functionality Testing:** Manual testing will be the primary method for verifying the software's core functionalities. This includes testing:
 - Uploading and processing PDF documents.
 - Flashcard creation based on extracted information or highlighted text passages.
- **Basic Usability Testing:** Informal usability testing will be conducted with a small group of users to identify any major usability issues that impede core tasks. Techniques like walkthroughs and user feedback sessions can be employed.

6.3 Evaluation Plan

Due to the project's scope, a formal evaluation involving a large user base might not be feasible. However, the following methods can be used to gather initial feedback and evaluate the software's effectiveness:

- **User Feedback Sessions:** Conducting user feedback sessions with participants from the usability testing can provide valuable insights into the software's ease of use, efficiency for creating flashcards, and overall user experience.
- **Self-Assessment Surveys:** Distributing short surveys to users after interacting with the software can help assess their perception of the software's usefulness, time-saving capabilities, and potential for improving their study habits.

6.4 Chapter Summary

This chapter provided an overview of the Flashcard Generation Software's implementation process, the testing methodologies employed to ensure functionality and usability, and the evaluation plan for gathering initial user feedback. The implemented core functionalities establish a foundation for the software, and the testing and evaluation activities provide valuable insights for future development and potential improvements.

07: Conclusions, Findings, and Recommendations

This chapter summarises the key takeaways from the development of the Flashcard Generation Software project. It highlights the project's achievements, discusses encountered challenges, and presents recommendations for future improvements and potential applications.

7.1 Introduction

The Flashcard Generation Software project aimed to develop a user-friendly software application to assist students in creating flashcards from PDFs. This software automates the process of information extraction and flashcard creation, saving students time and effort compared to manual methods.

7.2 Conclusions

The Flashcard Generation Software successfully established a foundation for automating flashcard creation from PDFs. The framework for uploading and processing PDFs was explored. While the initial focus was on user experience and core functionalities due to time constraints, the groundwork was laid for future exploration of information extraction techniques.

7.3 Challenges Encountered

- **Project Management:** Due to the project's scope and time constraints, full implementation of all envisioned features proved challenging. This highlights the importance of prioritising functionalities and establishing a realistic project timeline in future endeavours.
- **Information Extraction (NLP):** Exploring Natural Language Processing (NLP) techniques for information extraction was identified as a valuable future direction. However, due to the complexity of NLP and the project's timeframe, in-depth exploration was not possible within this iteration.

7.4 Future Recommendations

Building upon the project's foundation, here are some recommendations for future development:

- **Functional Enhancements:** Consider incorporating functionalities like [e.g., the option to export flashcards in different formats, the ability to edit or add notes to flashcards].
- **Advanced Features:** A key area for future exploration is information extraction using NLP techniques. This could enable features like [e.g., automatic identification of key

concepts within PDFs, ability to summarise extracted information for flashcard creation].

- **Usability Improvements:** Based on user testing (if conducted), identify areas for improvement in the user interface or user experience (e.g., implementing a tutorial for new users, or providing more visual options for flashcard customisation).

7.5 Conclusion

The Flashcard Generation Software successfully established a foundational framework for a user-friendly application to create flashcards from PDFs. While time constraints limited the implementation of information extraction functionalities using NLP, the project lays the groundwork for future exploration in this area. The project highlights the importance of effective project management and prioritising functionalities within a realistic timeframe.

References

Pyc, M. A., & Rawson, K. A. (2011). Improving educational software design: Does aligning cognitive load with student expertise lead to greater learning? *Learning and Instruction*, 21(2), 185-193. <https://www.sciencedirect.com/science/article/abs/pii/S0360131511002399>

Gasser, M., Schwartz, S. H., & Stevens, A. A. (2005). The influence of type fonts on information recall. *Perception*, 34(11), 1421-1432. https://www.researchgate.net/publication/237229931_The_Influence_of_Font_Type_on_

Open-Spaced Repetition. (n.d.). Anki flashcards with spaced repetition [FSRS4Anki]. GitHub Wiki. <https://github.com/open-spaced-repetition/fsrs4anki/wiki>

Mittal, N., Wu, S., Liu, Z., & Li, Y. (2022, May). A multimodal summarization framework for educational videos using transformers. In *Proceedings of the 2022 ACM Conference on Computer and Information Technologies* (pp. 1823-1832). Association for Computing Machinery. <https://doi.org/10.1145/3534678.3539081>