COMSATS University Islamabad,

Abbottabad Campus

SOFTWARE REQUIREMENTS SPECIFICATION
(SRS DOCUMENT)

for

Smart Editor
Version 1.0

By

Ahmed Haroon CIIT/FA19-BCS-018/ATD

Zohaib Nadeem CIIT/FA19-BCS-028/ATD

Hussain Ahmed CIIT/FA19-BCS-074/ATD

Supervisor
Ms. Humaira Jabeen

Bachelor of Science in Computer Science (2019-2022)

Contents

Revision History

Name Date Reason for changes Version

Application Evaluation History

Comments (by committee)

Action Taken

***include the ones given at scope time both in doc and presentation**

Supervised by

Humaira Jabeen

Signature_____

Introduction

Purpose

Identify the product or application whose requirements are specified in this document.

Scope

- Editing text and images: The ability to add, delete, and modify text and images within a PDF document.
- Adding and deleting pages: The ability to insert and delete pages within a PDF document.
- Splitting and merging documents: The ability to split a PDF document into multiple smaller documents, or merge multiple PDF documents into a single document.
- Annotation and marking up: The ability to add notes, comments, highlights, and other annotations to a PDF document.
- Conversion: The ability to convert a PDF document into other file formats, such as Microsoft Word or Excel, or vice versa.
- Security: The ability to password-protect PDF documents, or to add digital signatures to verify the authenticity of a document.

Overall, the scope of this Smart Editor app is to edit a PDF Document by converting it to word document. Here the document will be editable without losing the original content of the pdf document. A PDF document uses multiple compression algorithms, which include LZW77, or Huffman encoding algorithm. These compression algorithms are lossless algorithms. As PDF document are viewer friendly, and used to protect the original content being edited, they cannot be edited dynamically using any applications.

Overall description

Product perspective

From a product perspective, a PDF editor application is a software tool that allows users to create, edit, and modify PDF documents. It typically provides a range of features such as the ability to add, delete, or modify text and images within the document, as well as the ability to add, delete, or rearrange pages.

Some PDF editor applications may also offer advanced features such as the ability to annotate the document with notes or highlights, the ability to export the document to other file formats, and the ability to print the document.

One of the key benefits of a PDF editor application is the ability to make changes to a PDF document without the need for the original source file or specialized software. This can be particularly useful for editing documents that have been shared or received via email, as it allows users to make changes and share the updated document easily.

Overall, a PDF editor application can be a valuable tool for individuals and organizations looking to edit and modify PDF documents, and can help to improve efficiency and productivity by enabling users to make changes to documents quickly and easily.

- Text editing: This allows users to modify the text within a PDF document, including adding, deleting, and formatting text.
- Image editing: This allows users to add, delete, or modify images within a PDF document.
- Page manipulation: This allows users to add, delete, or rearrange pages within a PDF document.
- Annotation: This allows users to add notes, highlights, or other types of annotations to a PDF document.
- File conversion: This allows users to export a PDF document to other file formats, such as Word or Excel.
- Printing: This allows users to print a PDF document.
- Collaboration: This allows multiple users to edit and annotate a PDF document simultaneously.
- Security: This includes features such as password protection and digital signatures to ensure the security and integrity of the PDF document.
- Compatibility: This ensures that the PDF editor application can open and edit PDF documents created by other software.
- User interface: This refers to the design and layout of the application, which should be intuitive and easy to use.

Operating environment

Design and implementation constraints

There are several design and implementation constraints to consider when developing a PDF editor mobile application.

One constraint is the limited screen size and resolution of mobile devices, which can impact the user interface (UI) and user experience (UX) of the application. To address this constraint, the UI and UX design should be optimized for small screens and should consider the available screen real estate when displaying content.

Another constraint is the limited processing power and memory of mobile devices, which can impact the performance of the application. To address this constraint, the application should be designed to be lightweight and efficient, and should be optimized for the specific hardware and operating system of the target mobile devices.

A third constraint is the varying connectivity and network conditions of mobile devices, which can impact the ability to access and download content. To address this constraint, the application should be designed to handle offline scenarios and to gracefully handle interruptions in connectivity.

A fourth constraint is the need to support a range of mobile devices and operating systems, which can require the development of multiple versions of the application. To address this constraint, the application should be designed to be flexible and adaptable, and should use responsive design techniques to ensure a consistent experience across devices.

Overall, the design and implementation of a PDF editor mobile application should consider these and other constraints in order to deliver a high-quality and reliable user experience.

Requirement identifying technique

Use case diagram

Create a use case diagram using **MS Visio** for your system. For detail guideline to develop use case diagram, follow any of latest **UML book**]

Use case description

The table below indicate a comprehensive use case template filled in with an example drawn from the Cafeteria ordering system (COS). (Appendix C) shows more sample use cases written according to this template. As with all templates, you don't complete this from top to bottom, and you don't necessarily need all the template information for every use case. The template is simply a structure in which to store the information you encounter during a use case discussion in an organized and consistent fashion. The template reminds you of all the information you should contemplate regarding each use case. For more detail see Chapter 8, "Understanding user requirements"

Use Case	Description
Open PDF	The user selects a PDF file to open in the editor.
Edit Text	The user modifies the text within the PDF document.
Add Image	The user adds an image to the PDF document.
Delete Page	The user removes a page from the PDF document.
Split PDF	The user splits the PDF document into multiple documents.
Merge PDF	The user merges multiple PDF documents into a single document.
Annotate	The user adds annotations, such as notes or highlights, to the PDF document.
Export PDF	The user exports the edited PDF document in a desired format, such as .pdf or .docx.

Table 1 Show the detail use case template

Use Case ID:	**Enter a unique numeric identifier for the Use Case. e.g. UC-1**
Use Case Name:	**Enter a short name for the Use Case using an active verb phrase. e.g. Order a Meal**
Actors:	• **Admin: The administrator has full access to the system and can manage user accounts, set permissions, and monitor usage.** • **Regular users: These are the users who have access to the PDF editor and can create, edit, and convert PDF documents.** • **External parties: These are individuals or organizations that may interact with the PDF editor application, such as clients or partners. They may be able to view or edit PDF documents depending on the permissions set by the administrator.** • **System: The PDF document editor application itself, which handles the creation, editing, and conversion of PDF documents.** • **Hardware: The client computers and server that host and run the PDF document editor application.**
Description:	
Trigger:	**[Identify the event that initiates the use case.]e.g. A Patron indicates that he wants to order a meal.**
Preconditions:	**[List any activities that must take place, or any conditions that must be true, before the use case can be started.** **PRE-1. Patron is logged into COS.** **PRE-2. Patron is registered for meal payments by payroll deduction.**
Postconditions:	**[Describe the state of the system at the conclusion of the use case execution.** **POST-1. Meal order is stored in COS with a status of "Accepted."** **POST-2. Inventory of available food items is updated to reflect items in this order.** **POST-3. Remaining delivery capacity for the requested time window is updated.**
	[Provide a detailed description of the user actions and system responses that will take place during execution of the use case under normal, expected conditions. **1.0 Order a Single Meal** **1. Patron asks to view menu for a specific date. (see 1.0. E1, 1.0.E2)** **2. COS displays menu of available food items and the daily special.** **3. Patron selects one or more food items from menu. (see 1.1)** **4. Patron indicates that meal order is complete. (see 1.2)**

Normal Flow:

5. COS displays ordered menu items, individual prices, and total price, including taxes and delivery charge.

6. Patron either confirms meal order (continue normal flow) or requests to modify meal order (return to step 2).

7. COS displays available delivery times for the delivery date.

8. Patron selects a delivery time and specifies the delivery location.

9. Patron specifies payment method.

10. COS confirms acceptance of the order.

11. COS sends Patron an email message confirming order details, price, and delivery instructions.

12. COS stores order, sends food item information to Cafeteria Inventory System, and updates available delivery times.

[Document legitimate branches from the main flow to handle special conditions (also known as extensions). For each alternative flow reference the branching step number of the normal flow and the condition which must be true for this extension to be executed. e.g.

Alternative Flows:

[Alternative Flow 1 – Not in Network]

1.1 Order multiple identical meals

1. Patron requests a specified number of identical meals. (see 1.1. E1)

2. Return to step 4 of normal flow.

1.2 Order multiple meals

1. Patron asks to order another meal.

2. Return to step 1 of normal flow.

Note: Insert a new row for each distinctive alternative flow.]

Exceptions:

1.0. E1 Requested date is today and current time is after today's order cutoff time

1. COS informs Patron that it's too late to place an order for today.

2a. If Patron cancels the meal ordering process, then COS terminates use case.

2b. Else if Patron requests another date, then COS restarts use case.

1.0. E2 No delivery times left

1. COS informs Patron that no delivery times are available for the meal date.

2a. If Patron cancels the meal ordering process, then COS terminates use case.

2b. Else if Patron requests to pick the order up at the cafeteria, then continue with normal flow, but skip steps 7 and 8.

1.1. E1 Insufficient inventory to fulfill multiple meal order

1. COS informs Patron of the maximum number of identical meals he can order, based on current available inventory.

2a. If Patron modifies number of meals ordered, then return to step 4 of normal flow.

2b. Else if Patron cancels the meal ordering process, then COS terminates use case.

Business Rules

- **Users must have a valid account and be logged in to access the PDF editor.**
- **Users are only allowed to edit and modify PDF documents that they have permission to access.**
- **Changes made to a PDF document must be saved before the user can log out or close the application.**
- **If a user attempts to import a PDF document that is larger than the maximum file size allowed by the system, an error message will be displayed and the import will not be allowed.**
- **If a user attempts to export a PDF document in a format that is not supported by the system, an error message will be displayed and the export will not be allowed.**
- **If a user attempts to print a PDF document that is larger than the maximum number of pages allowed by the system, an error message will be displayed and the print will not be allowed.**
- **If a user attempts to sign a PDF document without a valid electronic signature, an error message will be displayed and the sign will not be allowed.**

Assumptions:

[List any assumptions.

1. **e.g. Assume that 15 percent of Patrons will order the daily special (Source: previous 6 months of cafeteria data).**

Functional Requirements

- View and edit PDF files: The application should be able to open, view, and edit PDF documents. This may include the ability to add, delete, or modify text and images, as well as to rearrange or delete pages within the document.
- Annotate PDF files: The application should allow users to add annotations to PDF documents, such as highlighting, underlining, or adding notes or comments.
- Convert PDF files: The application should be able to convert PDF documents to other file formats, such as Word or Excel, and vice versa.
- Merge and split PDF files: The application should allow users to merge multiple PDF documents into a single file, or split a single PDF document into multiple files.
- Protect PDF files: The application should provide features to password-protect PDF documents, as well as to restrict editing or printing.
- Customize PDF files: The application should allow users to customize the appearance of PDF documents, such as by adding watermarks or adjusting the layout.
- Support for various PDF versions: The application should support different versions of the PDF specification, such as PDF 1.7 and PDF 2.0.
- User-friendly interface: The application should have a user-friendly interface that is easy to navigate and use. It should also have features such as search and find, as well as the ability to bookmark pages and add notes.

Functional Requirement X

Itemize the specific functional requirements associated with each feature. These are the software capabilities that must be implemented for the user to carry out the feature's services or to perform a use case. Describe how the product should respond to anticipated error conditions and to invalid inputs and actions. Uniquely label each functional requirement, as described earlier. You can create multiple attributes for each functional requirement, such as rationale, source, dependencies etc. The following template is required to write functional requirements. For further detail see Chapter 11" Writing excellent requirements".

Table 2 Show the functional requirement template

Identifier	**Requirement ID**
Title	**Title of requirement**
Requirement	**Description of requirement which may be written either from user or system perspective e.g.** **If written in user perspective** **The [user class or actor name] shall be able to [do something] [to some object] [qualifying conditions, response time, or quality statement].** **If written in system perspective** **[optional precondition] [optional trigger event] the system shall [expected system response]**
Source	**Where this requirement is come from (who originate it)**
Rationale	**Motivation behind the requirement**
Business Rule (if required)	**Any restriction, policy, rule that the particular requirement must be fulfilled through its functional behavior**
Dependencies	**Requirements ID that are dependent on this requirement**
Priority	**High/Medium/Low**

Non-Functional Requirements

- Usability: The application should be easy to use and understand, with a clear and intuitive interface.
- Performance: The application should open, save, and manipulate PDF files efficiently, with minimal delays or errors.
- Compatibility: The application should support a wide range of PDF versions and file formats, and be able to import and export documents with minimal loss of data.
- Security: The application should protect the privacy and security of user data, and prevent unauthorized access to or manipulation of PDF documents.
- Reliability: The application should be stable and reliable, with minimal crashes or errors.
- Scalability: The application should be able to handle large PDF files and a large number of users without degrading performance.
- Maintainability: The application should be easy to maintain and update, with a clear and modular design.
- Internationalization: The application should support multiple languages and localization options.
- Accessibility: The application should be accessible to users with disabilities, and comply with relevant accessibility standards.

.

References

List any documents or other resources to which this SRS refers, if any. These might include user interface style guides, standards, system requirements specifications, interface specifications, or the SRS for a related product.