

**K. J. Somaiya School of Engineering**  
**Department of Computer Engineering**

**Batch: A-4    Roll No.: 16010122151**

**Experiment No. 04**

**Group No :- 5**

**Signature of the Staff In-charge with date**

**Title: Design of Test Cases**

---

**Expected Outcome of Experiment:**

**CO2: Implement and prototype creation for the specified application.**

---

**Books/ Journals/ Websites referred:**

*[Students can mention websites/ books used in their project implementation]*

---

**This write up will expect students to prepare chapter no. 4 in the format given below**

## Chapter 4

### Design Test Cases

*This chapter details the testing conducted mentioning the various test scenarios considered.* This chapter in a software development project report should provide a comprehensive overview of the testing process, methodologies, and results. The purpose of a test case document is to provide a detailed, structured record of the steps and conditions needed to verify a specific functionality or feature in software testing, ensuring that the application functions correctly and meets requirements

#### **Introduction:**

Developing design test cases is crucial for identifying defects early, ensuring software quality, and validating user requirements, ultimately leading to a more reliable and user-friendly product.

Testing of a software system is a very essential step which provides benefits such as:

- **Early Defect Detection:**

Thoroughly designed test cases help uncover defects early in the development lifecycle, minimizing costly rework and delays.

- **Enhanced Software Quality:**

By systematically testing all features and functionalities, test cases ensure that the software meets user expectations and performs as intended.

- **Improved User Satisfaction:**

A well-tested product with fewer bugs leads to a better user experience and increased satisfaction.

- **Reduced Costs:**

Identifying and fixing defects early in the development process is significantly cheaper than fixing them later, saving time and resources.

- **Validation of Requirements:**

Test cases verify that the software meets the defined requirements and functionalities, ensuring that the product delivers on its promises.

- **Increased Confidence:**

Thorough testing builds confidence among stakeholders, including customers, investors, and regulatory bodies, regarding the software's quality and reliability.

- **Optimized Testing Effort:**

Well-designed test cases ensure comprehensive test coverage, minimizing redundancy and maximizing efficiency.

- **Facilitates Bug Detection:**

Test cases help identify when something in the software isn't working correctly, allowing bugs to be fixed before the software is released.

- **Ensures Coverage:**

They help to ensure that every feature of the software is tested and that all the requirements are covered.

- **Traceability:**

Test cases can be traced back to the requirements they are testing, providing a clear link between the requirements and the testing process.

- **Test Scenarios:**

Test scenarios are useful for organizing your test cases and providing high-level information on what to test in your application.

- **Test Coverage:**

Test coverage reports on areas of the codebase that are not covered by the test cases, facilitating the creation of new test cases to ensure maximum test coverage.

Students will be required to perform tests (whichever are applicable) from the below list:

1. **Validation of data:**

Checking if the entered values are in the correct format and within the range etc.

2. **Appropriate Navigation**

Accessing the various pages as per the flow of the system

3. **Verification of the results generated using the system.**

Check if the results are correct

### Introduction

Testing plays a critical role in verifying that the "Chat with PDF" AI model performs as expected—accurately parsing, understanding, and responding to queries about the content of uploaded PDF files. The test cases designed for this project validate file inputs, system behavior, and output accuracy, ensuring the model meets both user needs and project requirements.

Benefits of this testing phase include:

- **Early Defect Detection:** Identifies bugs in PDF parsing, context extraction, and response generation early in development.
- **Improved Software Quality:** Ensures the system behaves reliably across different file types and query formats.
- **Enhanced User Satisfaction:** Accurate, quick, and contextually relevant responses increase user trust and engagement.
- **Validation of Requirements:** Confirms the system meets all functional objectives, including PDF handling and interactive chat capability.

### Test Scenarios and Test Cases

#### 1. Validation of PDF Input

Test Case ID	Description	Input	Expected Output	Result
TC_01	Empty PDF upload	(No file)	Show "Please upload a PDF file"	Pass
TC_02	Non-PDF file format	.docx, .jpg	Show "Invalid file format"	Pass
TC_03	Large PDF file	> 50MB PDF	Accept and process or show performance warning	Pass
TC_04	Corrupted PDF	Malformed file	Show "Unable to read PDF content"	Pass

#### 2. Appropriate Navigation

Test Case ID	Description	Action	Expected Output	Result
TC_05	Load Home Page	Access root URL or app	UI loads successfully	Pass
TC_06	Upload PDF and start chat	Upload a valid PDF and enter a question	Redirect to chat interface with ready state	Pass
TC_07	Navigate back to upload	Click "Back" or "Home" button	Return to PDF upload section	Pass

#### 3. Verification of AI Responses

Test Case ID	Description	Input	Expected Output	Result
TC_08	Ask factual question	"What is the total revenue mentioned in the report?"	Accurate answer based on document content	Pass

Test Case ID	Description	Input	Expected Output	Result
TC_09	Ask contextual question	“Summarize the conclusion section.”	Concise and accurate summary	Pass
TC_10	Ask unrelated question	“What’s the weather today?”	Show "Query not related to uploaded document"	Pass
TC_11	Multi-turn interaction	Follow-up: “What about the next year?”	Context-aware response	Pass

### Test Coverage

This project covers:

- Validation of uploaded PDF files and format
- Smooth navigation from file upload to chat interface
- Verification of AI response accuracy
- Contextual follow-up and edge-case handling in queries

### Conclusion

The tests confirmed that the "Chat with PDF" system is stable, functional, and user-friendly. It effectively handles a variety of document types and user queries, supports intuitive navigation, and delivers context-aware, accurate responses. These tests enhance confidence in real-world deployment and showcase the project's reliability and usefulness.