Chatbot System - UML Documentation

# UML Diagrams (Images)

## Use Case Diagram

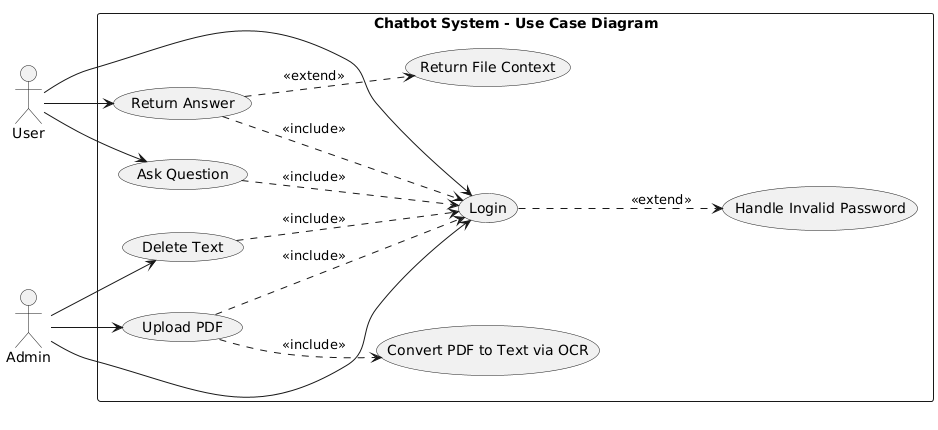


Figure: Use Case Diagram

## Login Sequence

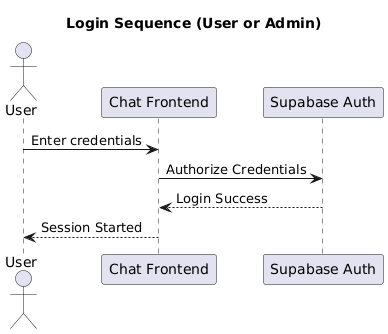


Figure: Login Sequence

## Handle Invalid Password Sequence

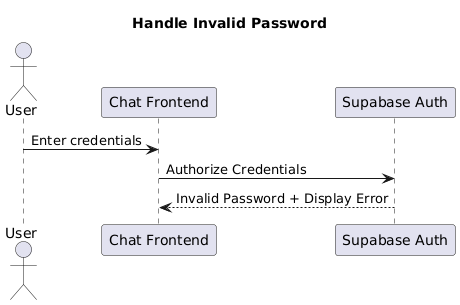


Figure: Handle Invalid Password Sequence

## Ask Question Sequence

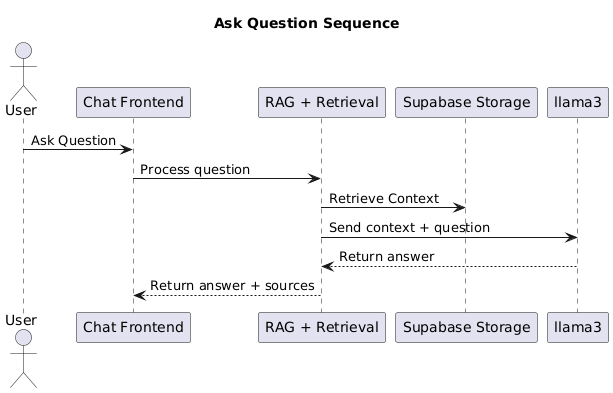


Figure: Ask Question Sequence

## Return File Context Sequence

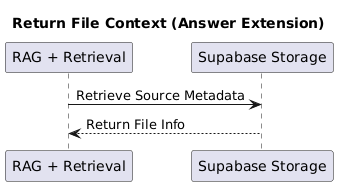


Figure: Return File Context Sequence

## Upload PDF Sequence

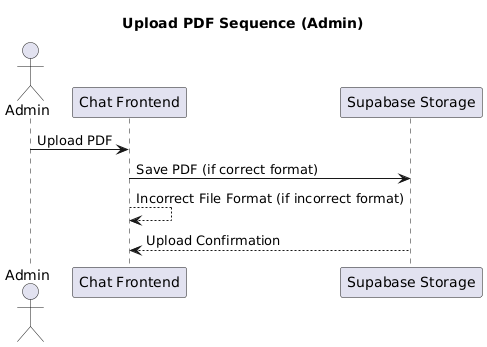


Figure: Upload PDF Sequence

## Convert PDF to Text Sequence

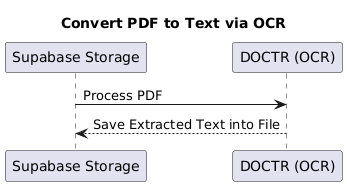


Figure: Convert PDF to Text Sequence

## Delete Text Sequence

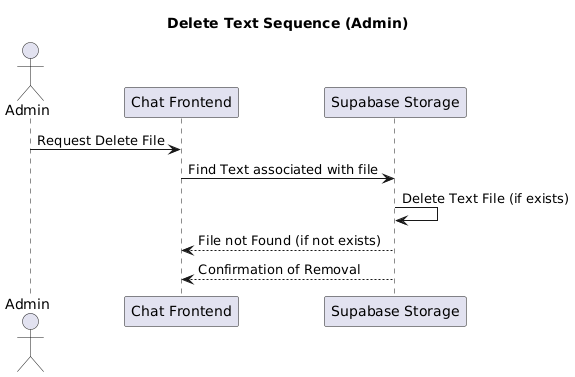


Figure: Delete Text Sequence

## UML Class Diagram

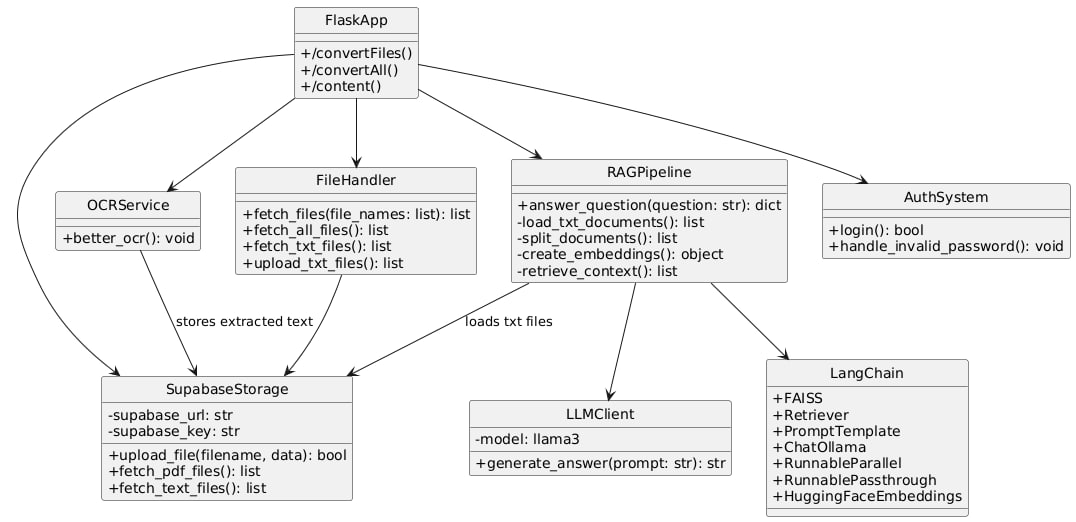


Figure: UML Class Diagram

## Components Diagram

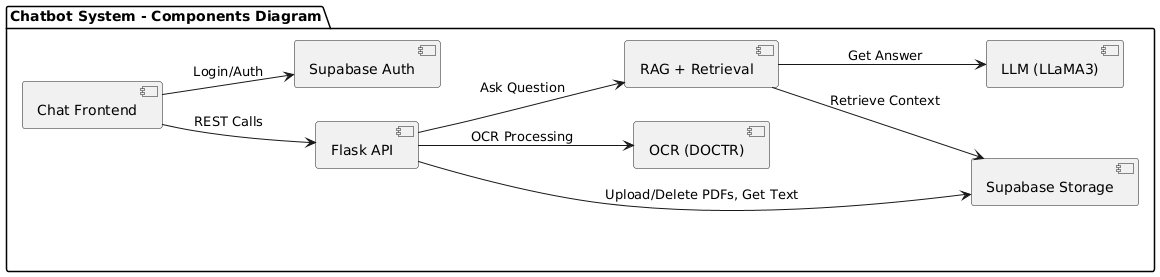


Figure: Components Diagram

# UML Diagrams (PlantUML Code)

## Use Case Diagram

-- Use Case Diagram --  
  
  
@startuml  
left to right direction  
  
actor User  
actor Admin  
  
rectangle "Chatbot System - Use Case Diagram" {  
 usecase "Login" as UC\_Login  
 usecase "Ask Question" as UC\_Ask  
 usecase "Return Answer" as UC\_Answer  
 usecase "Upload PDF" as UC\_Upload  
 usecase "Delete Text" as UC\_Delete  
 usecase "Convert PDF to Text via OCR" as UC\_Convert  
   
  
 ' Extended use cases  
 usecase "Handle Invalid Password" as UC\_BadLogin  
 usecase "Return File Context" as UC\_Context  
}  
  
' Actor connections  
User --> UC\_Ask  
User --> UC\_Answer  
User --> UC\_Login  
  
Admin --> UC\_Upload  
Admin --> UC\_Delete  
Admin --> UC\_Login  
  
' Includes (required behaviors)  
UC\_Ask ..> UC\_Login : <<include>>  
UC\_Answer ..> UC\_Login : <<include>>  
UC\_Upload ..> UC\_Login : <<include>>  
UC\_Delete ..> UC\_Login : <<include>>  
  
UC\_Upload ..> UC\_Convert : <<include>>  
  
' Extends (conditional behaviors)  
UC\_Login ..> UC\_BadLogin : <<extend>>  
UC\_Answer ..> UC\_Context : <<extend>>  
  
@enduml

## Sequence Diagrams

-- Login Sequence --  
  
@startuml  
title Login Sequence (User or Admin)  
actor User  
participant "Chat Frontend" as UI  
participant "Supabase Auth" as Auth  
  
User -> UI : Enter credentials  
UI -> Auth: Authorize Credentials  
Auth --> UI : Login Success  
UI --> User : Session Started  
@enduml  
  
-- Handle Invalid Password --  
  
@startuml  
title Handle Invalid Password  
actor User  
participant "Chat Frontend" as UI  
participant "Supabase Auth" as Auth  
  
User -> UI : Enter credentials  
UI -> Auth : Authorize Credentials  
Auth --> UI : Invalid Password + Display Error  
@enduml   
  
  
  
  
  
  
-- Ask Question Sequence --  
  
@startuml  
title Ask Question Sequence  
actor User  
participant "Chat Frontend" as API  
participant "RAG + Retrieval" as RAG  
participant "Supabase Storage" as Storage  
participant "llama3" as LLM  
  
User -> API : Ask Question  
API -> RAG : Process question  
RAG -> Storage: Retrieve Context  
RAG -> LLM : Send context + question  
LLM --> RAG : Return answer  
RAG --> API : Return answer + sources  
@enduml  
  
-- Return File Context (Answer Extension) --  
  
@startuml  
title Return File Context (Answer Extension)  
participant "RAG + Retrieval" as RAG  
participant "Supabase Storage" as Storage  
  
RAG -> Storage : Retrieve Source Metadata  
Storage --> RAG : Return File Info  
@enduml  
  
  
  
  
  
-- Upload PDF Sequence (Admin) --  
  
@startuml  
title Upload PDF Sequence (Admin)  
actor Admin  
participant "Chat Frontend" as UI  
participant "Supabase Storage" as Storage  
  
Admin -> UI : Upload PDF  
UI -> Storage : Save PDF (if correct format)  
UI --> UI : Incorrect File Format (if incorrect format)  
Storage --> UI : Upload Confirmation  
@enduml  
  
-- Convert PDF to Text via OCR --  
  
@startuml  
title Convert PDF to Text via OCR  
participant "Supabase Storage" as Storage  
participant "DOCTR (OCR)" as OCR  
  
Storage -> OCR : Process PDF  
OCR --> Storage: Save Extracted Text into File  
@enduml  
  
  
-- Delete Text Sequence (Admin) --  
  
@startuml  
title Delete Text Sequence (Admin)  
actor Admin  
participant "Chat Frontend" as UI  
participant "Supabase Storage" as Storage  
  
Admin -> UI : Request Delete File  
UI -> Storage : Find Text associated with file  
Storage -> Storage : Delete Text File (if exists)  
Storage --> UI : File not Found (if not exists)  
Storage --> UI : Confirmation of Removal  
@enduml

## Class Diagram

-- UML Class Diagram --  
  
@startuml  
skinparam classAttributeIconSize 0  
skinparam style strictuml  
  
' === Logical Service Classes Based on Actual Code ===  
  
class FlaskApp {  
 +/convertFiles()  
 +/convertAll()  
 +/content()  
}  
  
class SupabaseStorage {  
 -supabase\_url: str  
 -supabase\_key: str  
 +upload\_file(filename, data): bool  
 +fetch\_pdf\_files(): list  
 +fetch\_text\_files(): list  
}  
  
class OCRService {  
 +better\_ocr(): void  
}  
  
class FileHandler {  
 +fetch\_files(file\_names: list): list  
 +fetch\_all\_files(): list  
 +fetch\_txt\_files(): list  
 +upload\_txt\_files(): list  
}  
  
class RAGPipeline {  
 +answer\_question(question: str): dict  
 -load\_txt\_documents(): list  
 -split\_documents(): list  
 -create\_embeddings(): object  
 -retrieve\_context(): list  
}  
  
class LLMClient {  
 -model: llama3  
 +generate\_answer(prompt: str): str  
}  
  
class LangChain {  
 +FAISS  
 +Retriever  
 +PromptTemplate  
 +ChatOllama  
 +RunnableParallel  
 +RunnablePassthrough  
 +HuggingFaceEmbeddings  
}  
  
class AuthSystem {  
 +login(): bool  
 +handle\_invalid\_password(): void  
}  
  
' === Relationships ===  
FlaskApp --> FileHandler  
FlaskApp --> RAGPipeline  
FlaskApp --> OCRService  
FlaskApp --> SupabaseStorage  
FlaskApp --> AuthSystem  
  
FileHandler --> SupabaseStorage  
OCRService --> SupabaseStorage : stores extracted text  
RAGPipeline --> SupabaseStorage : loads txt files  
RAGPipeline --> LangChain  
RAGPipeline --> LLMClient  
  
@enduml

## Components Diagram

-- Components Diagram --  
  
@startuml  
left to right direction  
skinparam componentStyle uml2  
  
package "Chatbot System - Components Diagram" {  
  
component "Chat Frontend" as UI  
component "Flask API" as API  
component "Supabase Storage" as DB  
component "Supabase Auth" as AUTH  
component "OCR (DOCTR)" as OCR  
component "RAG + Retrieval" as RAG  
component "LLM (LLaMA3)" as LLM  
  
' === Associations ===  
UI --> API : REST Calls  
API --> DB : Upload/Delete PDFs, Get Text  
API --> OCR : OCR Processing  
API --> RAG : Ask Question  
RAG --> DB : Retrieve Context  
RAG --> LLM : Get Answer  
UI --> AUTH : Login/Auth  
}  
@enduml