

REST API for Document Viewing

Team 23

Aditya Chattopadhyay 201264201

Alok Thatikunta 201264197

Abhishek Saban 201405601

Outline

- ▶ Introduction
- ▶ Design
- ▶ Approach
- ▶ References

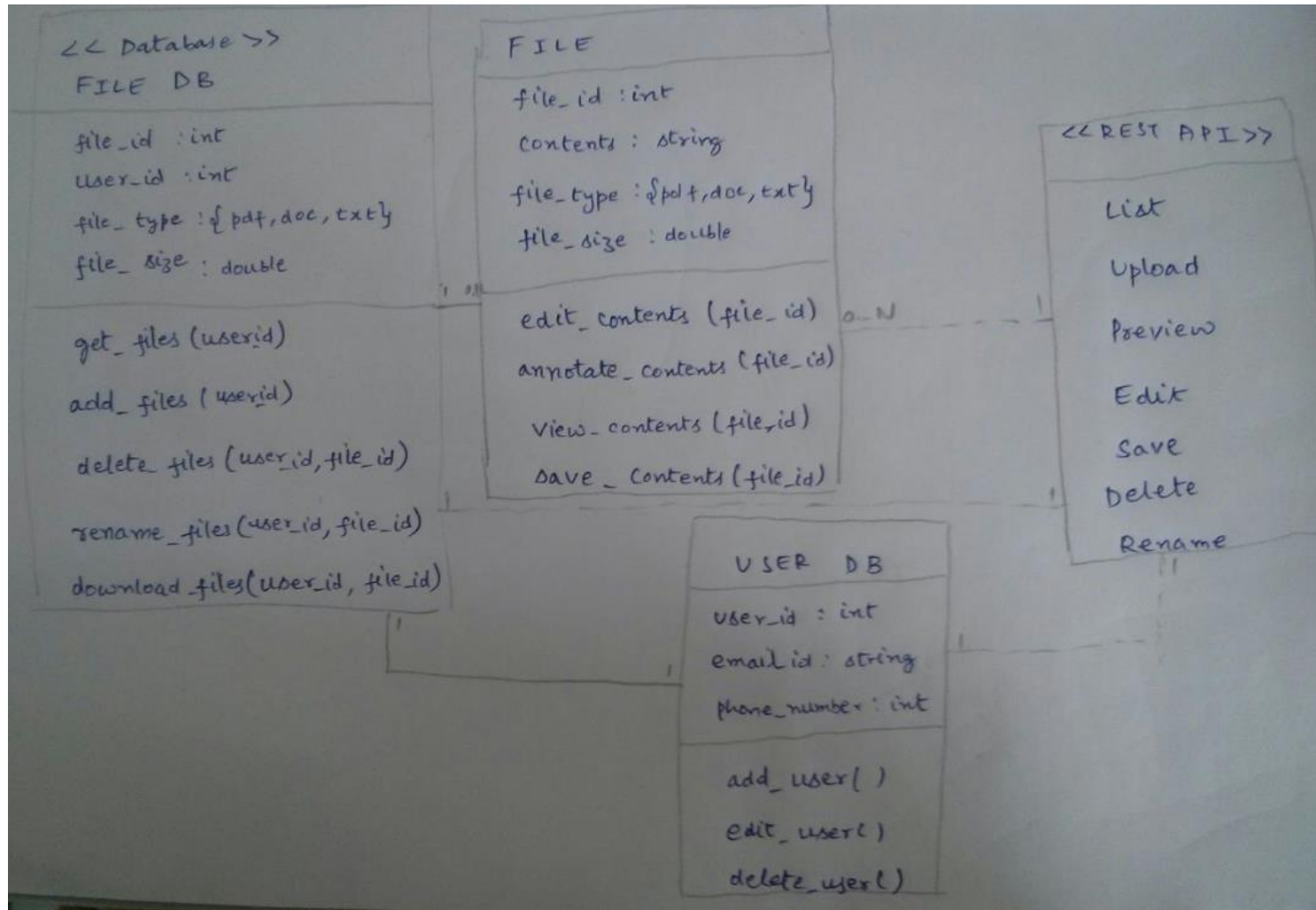
Introduction

- ▶ This Product allows you to access documents directly on our server and modify them!
- ▶ All functionalities are exposed via a REST-API allowing an efficient command-line interface for users to upload & manipulate their documents.
- ▶ This technology can be easily integrated with cloud providers for embedding documents and annotations on the fly, without the user having to explicitly download them, make changes and then re-upload.

Outline

- ▶ Introduction
- ▶ **Design**
- ▶ Approach
- ▶ References

Architectural Design



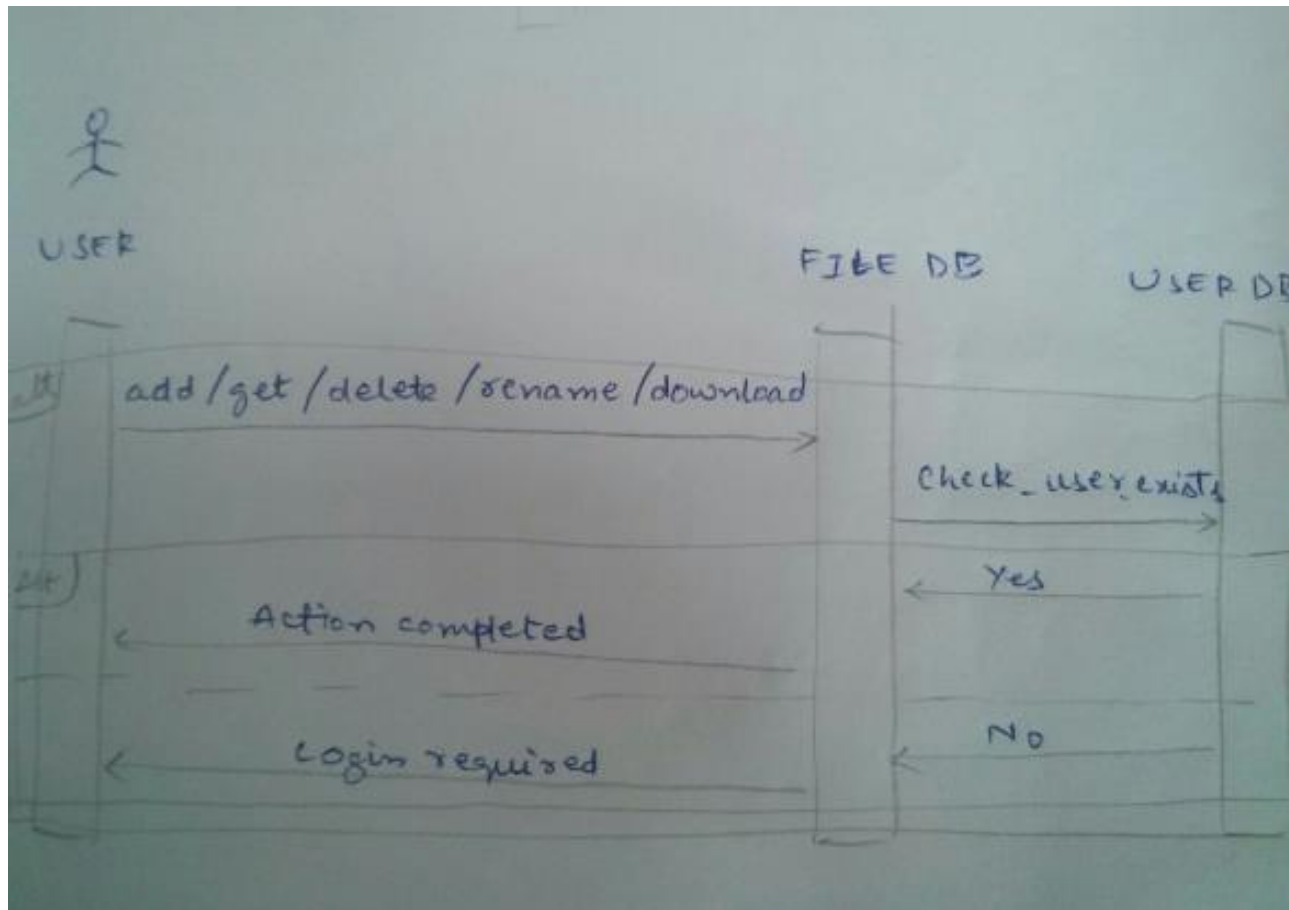
Sequence Diagrams of Functionalities

User Handling



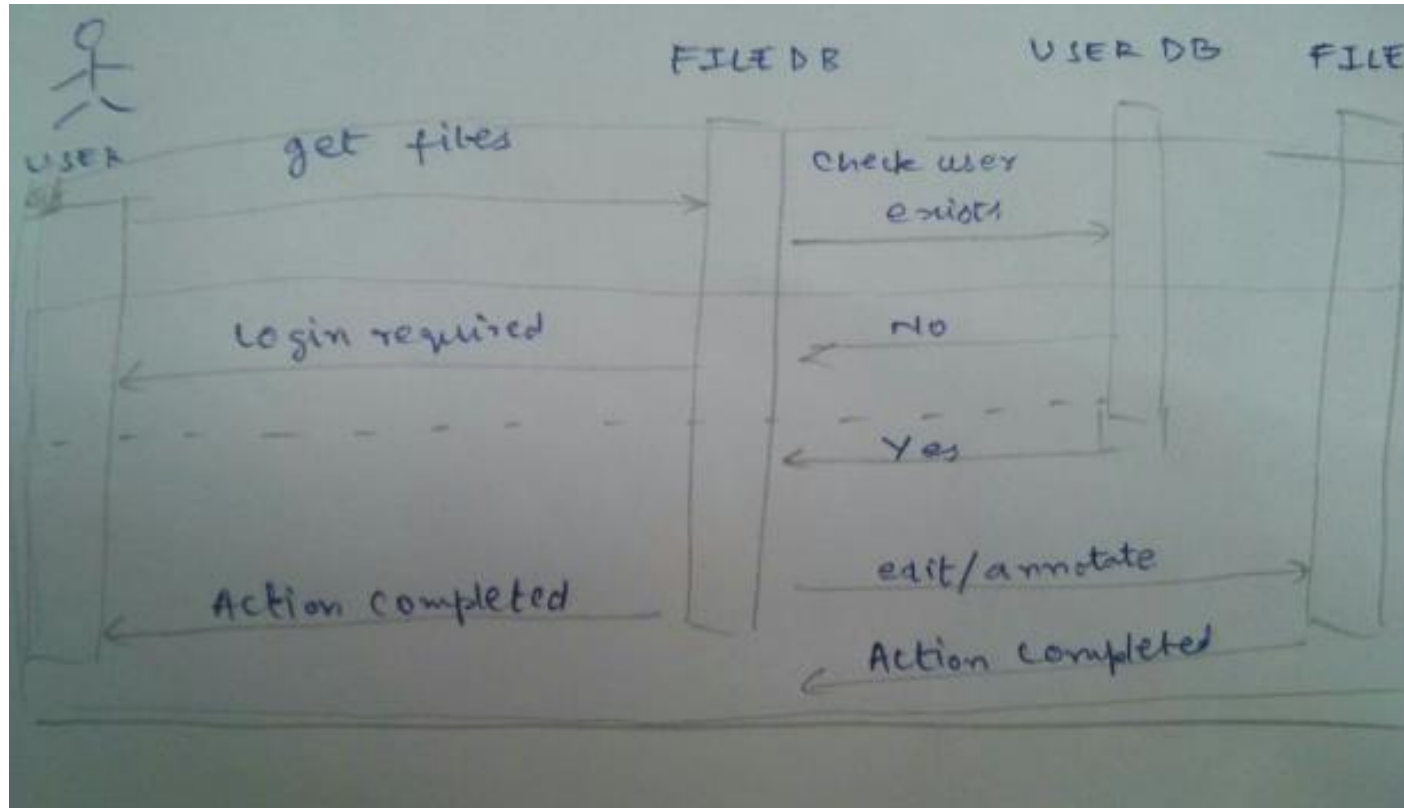
Sequence Diagrams of Functionalities

File Upload/Delete/Rename/Download



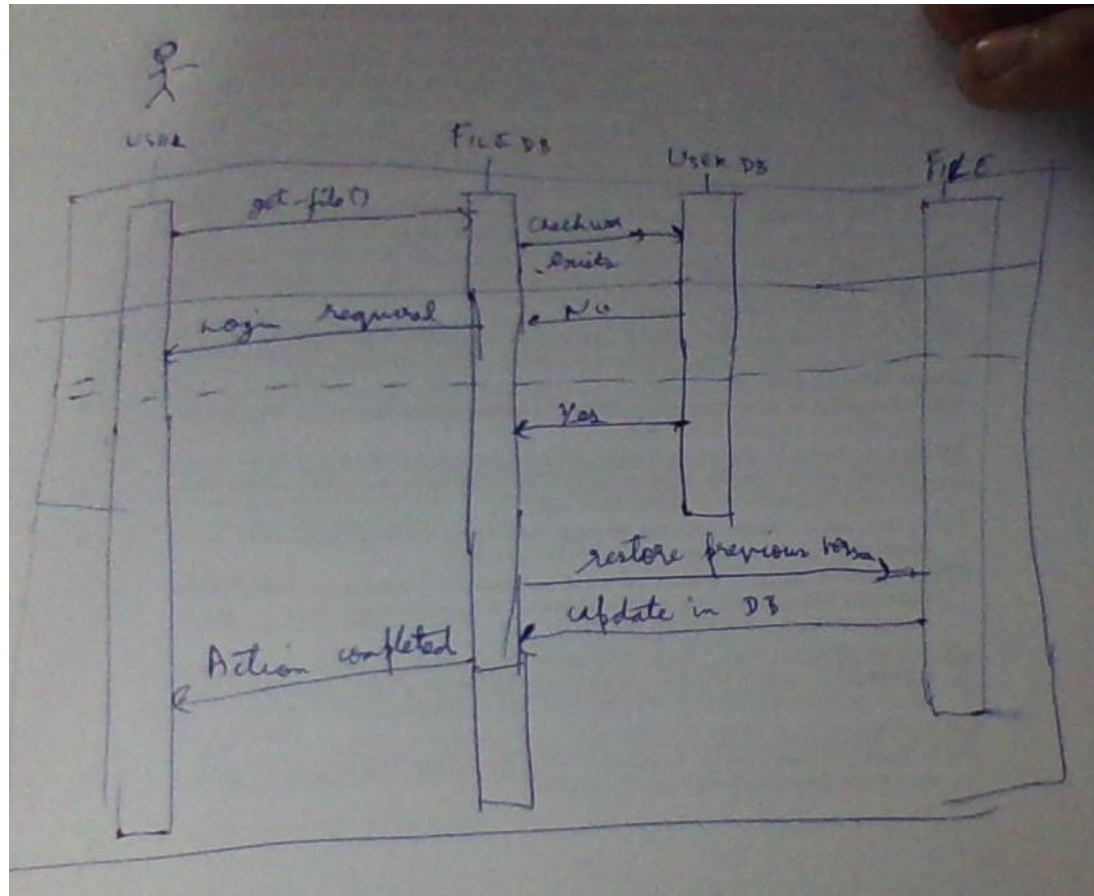
Sequence Diagrams of Functionalities

File Edit/Annotate



Sequence Diagrams of Functionalities

File Restore (Versioning)



The REST-API

- ▶ A REST-API which exposes each of the above described functionalities (via sequence diagrams).
- ▶ Each user can only access his/her own files (restricted user access).
- ▶ Two separate databases maintained - a User DB for registered users and a File DB which has a foreign key username from User DB.
- ▶ The user has to pass authentication tokens with each API call, to ensure that he/she can only access resources belonging to him/her.

Outline

- ▶ Introduction
- ▶ Design
- ▶ **Approach**
- ▶ References

Technologies Used

- ▶ Python based web framework (Flask) was used for rapid development.
- ▶ SQLite database was used for testing purposes this can be easily replaced with a more suited noSQL database or graph database (Neo4J).
- ▶ Other third-party python libraries used - python packages: flask, flask_wtf, wtforms, pdfminer, fdpf.
- ▶ Currently two types of files are supported - pdf & txt.
- ▶ Two modes of accessing the functionalities of our product - web interface & REST API

Functionalities

- ▶ **Register & Login** - Maintains a database of all the users and files associated with them.
- ▶ **Upload** - Users can upload pdf/text documents onto our servers.
- ▶ **View** - They can either directly view the uploaded document or a text version of it (through the REST-API).
- ▶ **List** - Lists all the files belonging to a user.
- ▶ **Delete** - Users can delete their files.
- ▶ **Edit** - Users can choose to edit documents online and save them without having to download them on their local machines. Currently pdf documents are converted to text, edited and saved as a pdf. Due to this formatting such as font-size, bold, italics etc. are lost!

Functionalities

- ▶ **Rename** - Users can rename files on our server, this is also updated in our file database which maps each file to its respective owner.
- ▶ **Restore** - Each time a user edits his/her file a backup copy of the previous version is stored on the server. The user can always restore the immediate previous version of their file in-case something goes wrong with their editing.

Outline

- ▶ Introduction
- ▶ Design
- ▶ Approach
- ▶ **References**

References

- ▶ Code and Documentation on GitHub -
https://github.com/darkwrath/Cloud_Docspad
- ▶ Slides on Slideshare -
<http://www.slideshare.net/adityachattopadhyay5/docspad-55388674>
- ▶ Demo Video on Youtube Channel -
<https://www.youtube.com/user/adityababiblue/videos>