SOFTWARE DESIGN

Project: The PDF evaluator

Team: TEAM 10

Revision History

DATE	VERSION	DESCRIPTION	AUTHOR
09-18-2023	1.o	Use case Description	Praveen Challa, Jaidithya Yarava
09-18-2023	1.o	Use Case Diagram	Jaidithya Yarava
09-20-2023	1.o	Functional Requirements	Aasritha Reddy Samanthula
09-20-2023	1.o	Non-functional Requirements	Aasritha Reddy Samanthula
09-20-2023	1.o	System Requirements	Tarun Teja Pasupuleti
09-20-2023	1.o	stakeholders	Tarun Teja Pasupuleti, Chandrika Vasantha
09-20-2023	1.o	Problem statement stakeholders	Chandrika Vasantha
09-21-2023	1.o	Individual Contributions	Praveen challa
09-24-2023	2.o	Class Diagram	Jaidithya Yarava, Tarun pasupuleti
09-24-2023	2.o	ER Diagram	AasrithaReddy, Jaidithya Yarava

09-24-2023	2.o	Sequence Diagram	Aasritha Reddy, Jaidithya yarava, Tarun Pasupuleti, Praveen Challa, Chandrika Vasantha
09-25-2023	2.o	Information Architecture diagram	Chandrika Vasantha
09-25-2023	2.o	Design Rationale	Aasritha Reddy, Tarun Pasupuleti, Chandrika Vasantha
09-25-2023	2.o	User Interface	Challa Praveen, Tarun Pasupuleti

Trello link : https://trello.com/b/pqbcBZ3a/sprint-2

Class Diagram

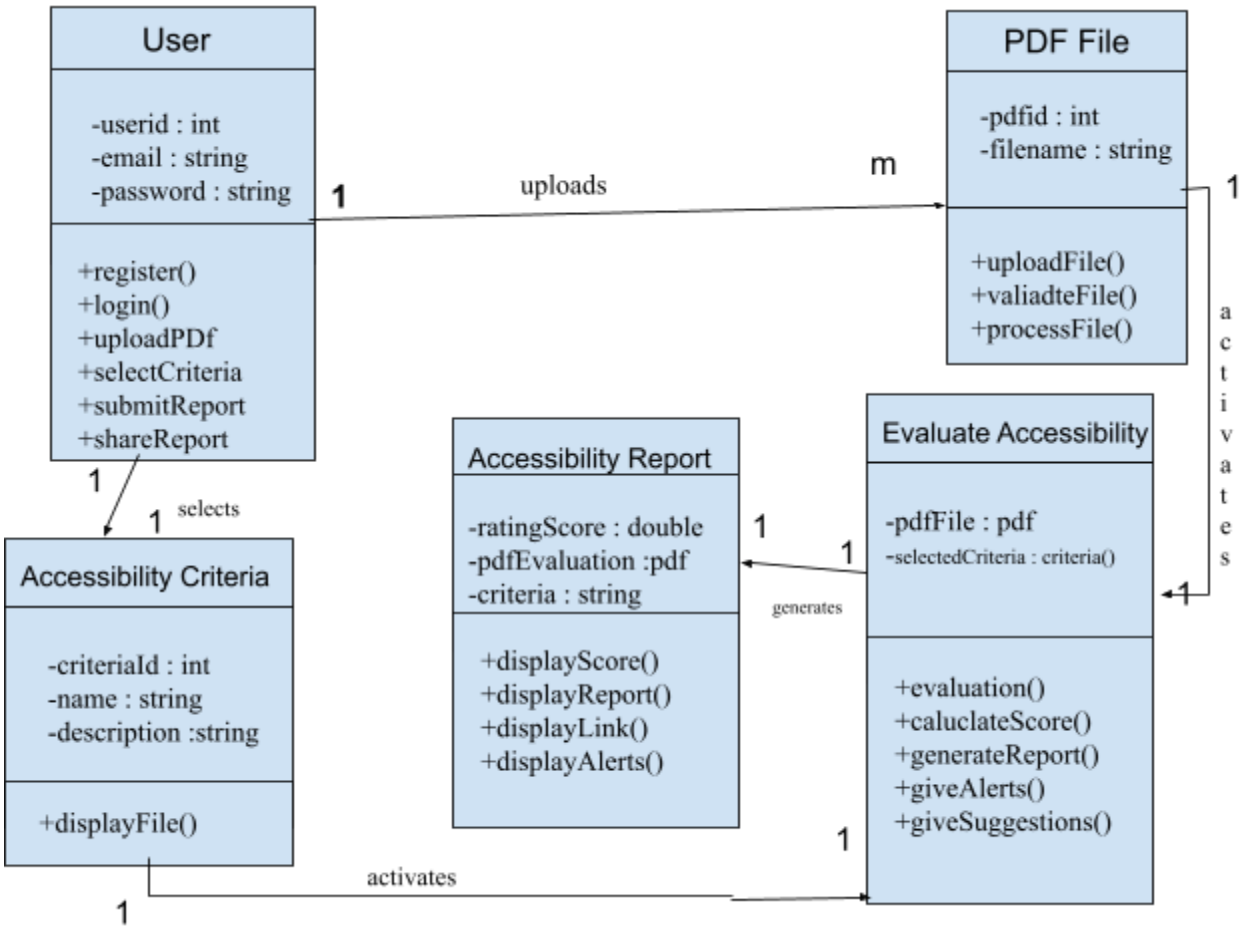
The **User class** represents registered users on the system. It contains attributes such as user ID, email, password, and functionalities, which represent the list of functionalities associated with the user. The User class has methods such as register(), login(), uploadPDF(), selectCriteria(), submitReport(), and shareReport().

The **PDF File class** represents a PDF file in the system. It has attributes such as pdfId and filename. The class provides methods like uploadFile(), validateFile(), and processFile().

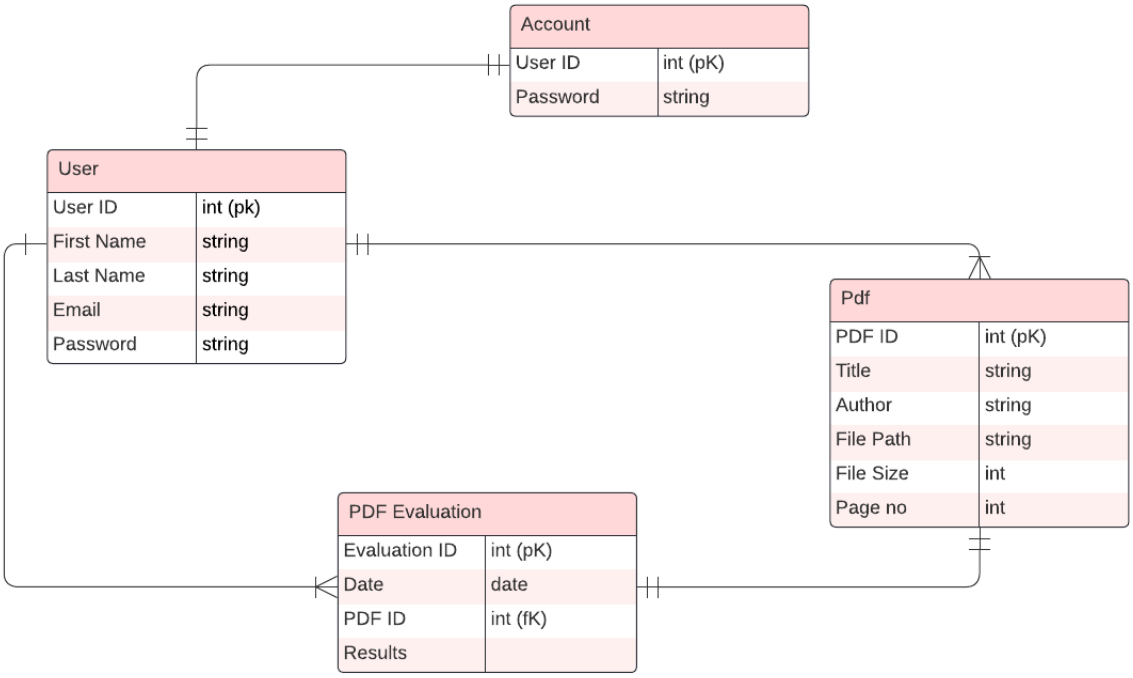
The **Accessibility Criteria** class represents the criteria used for evaluating accessibility. It has attributes such as criteria, name, and description. The class provides the functionality displayFile().

The **Evaluate Accessibility** class represents the evaluation process for accessibility. It has attributes like pdfFile and selectedCriteria. The class provides methods such as evaluation(), calculateScore(), generateReport(), giveAlerts(), and giveSuggestions().

The **Accessibility Report** class represents the generated report for accessibility evaluation. It has attributes like rating score, pdf evaluation, and criteria. The class provides functionalities such as displayScore(), displayReport(), displayLink(), and displayAlerts().


User
-userid : int
-email : string
-password : string
+register()
+login()
+uploadPDf
+selectCriteria
+submitReport
+shareReport
1
uploads
m
PDF File
-pdfid : int
-filename : string
+uploadFile()
+valiadteFile()
+processFile()
1
activates
1
selects
1
Accessibility Criteria
-criteriaId : int
-name : string
-description :string
+displayFile()
1
activates
Accessibility Report
-ratingScore : double
-pdfEvaluation :pdf
-criteria : string
+displayScore()
+displayReport()
+displayLink()
+displayAlerts()
1
1
generates
Evaluate Accessibility
-pdfFile : pdf
-selectedCriteria : criteria()
+evaluation()
+caluclateScore()
+generateReport()
+giveAlerts()
+giveSuggestions()
1

ER Diagram(s)



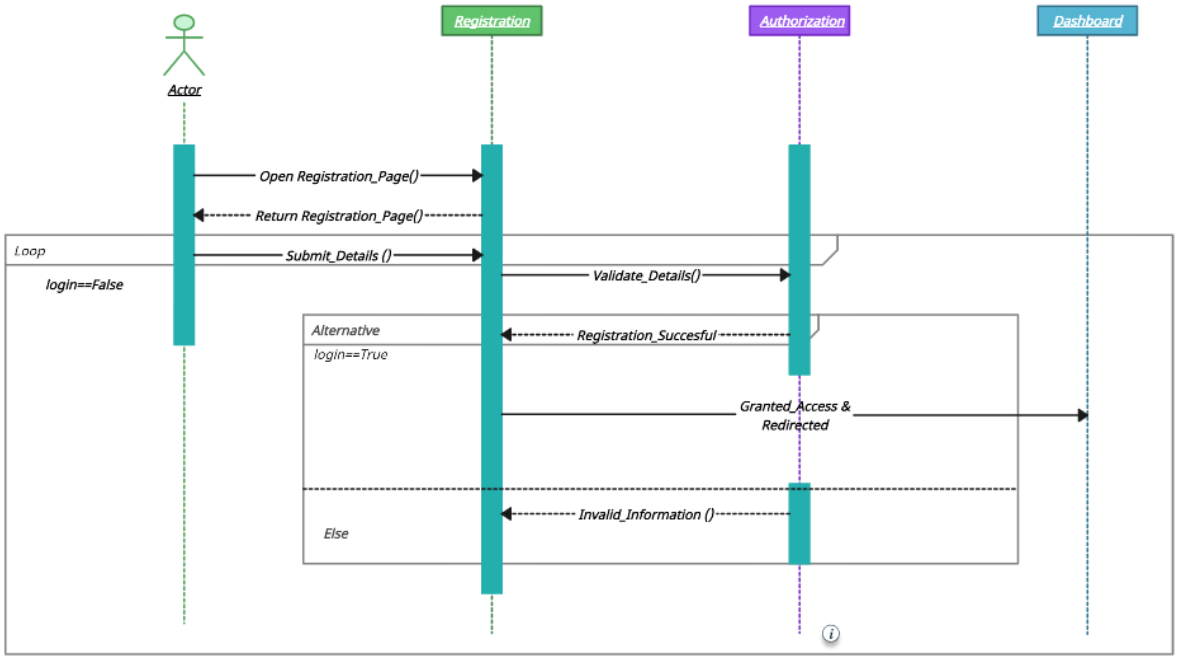
We don't need a database or records for future references, as per the project requirements. Therefore, the only two tables that our database really needs are login and registration, where a user can choose whether to log in or register. Whether he is a returning user or a first-time userwill depend.

Since a user can only register with one account once, there is a one-to-one relationship between the user table and the register table in this case. A person can log in as many times as they like, thus we also have a one-to-many relationship. A user can evaluate number of pdf's based on his/her interest. So, the relation between user and pdf evaluation entities is a one to many.

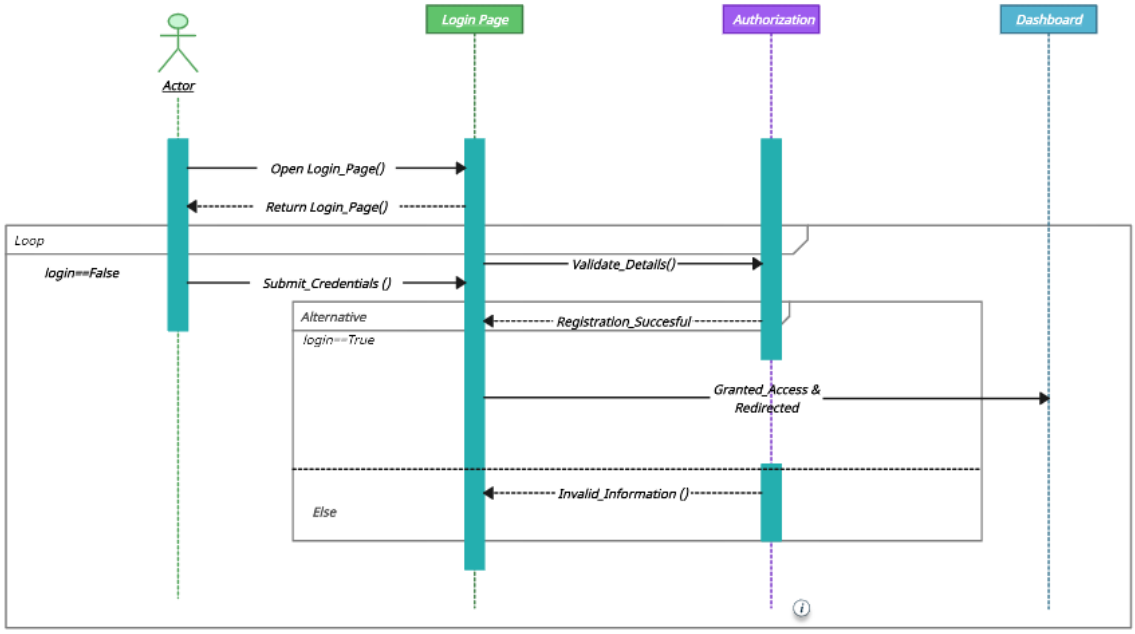
finally, the user can upload different number of pdf's for the evaluation. So, there is a one to one or many relation between user and pdf entities.

Sequence Diagram(s)

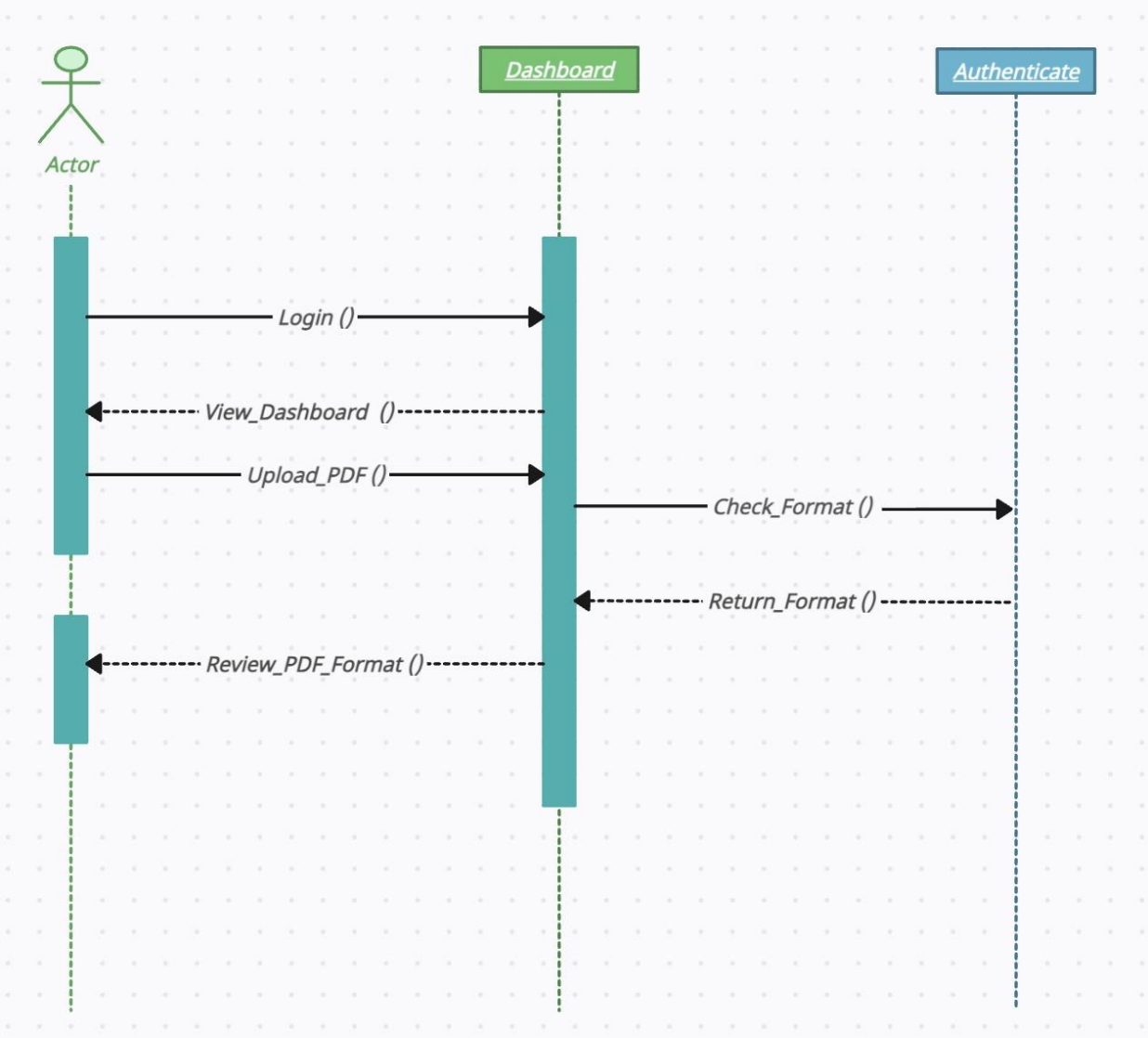
1. Registration Page :



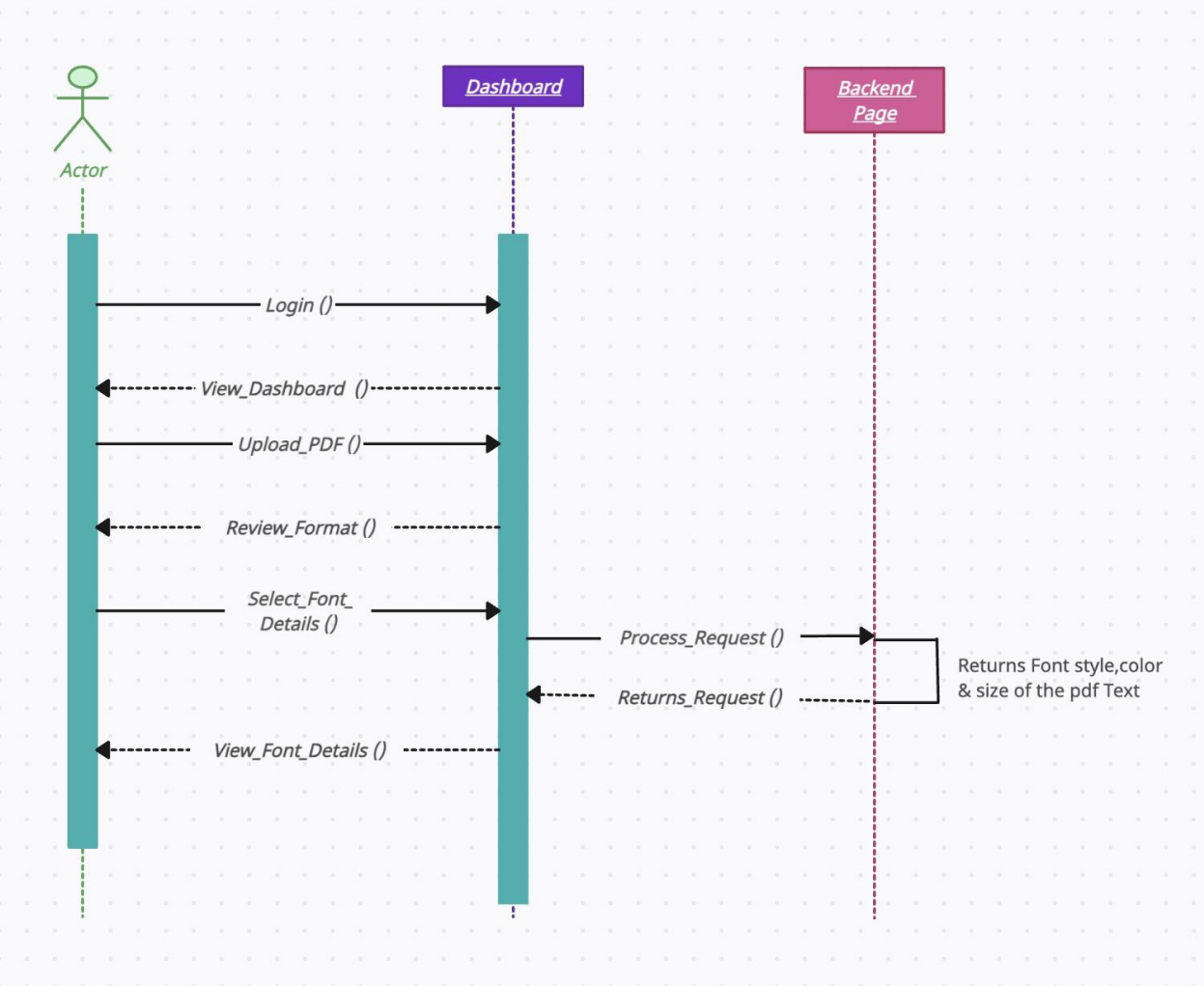
2.Login Page :


Actor
Login Page
Authorization
Dashboard
Open Login_Page()
Return Login_Page()
Loop
login==False
Submit_Credentials ()
Validate_Details()
Alternative
login==True
Registration_Succesful
Granted_Access &
Redirected
Else
Invalid_Information ()

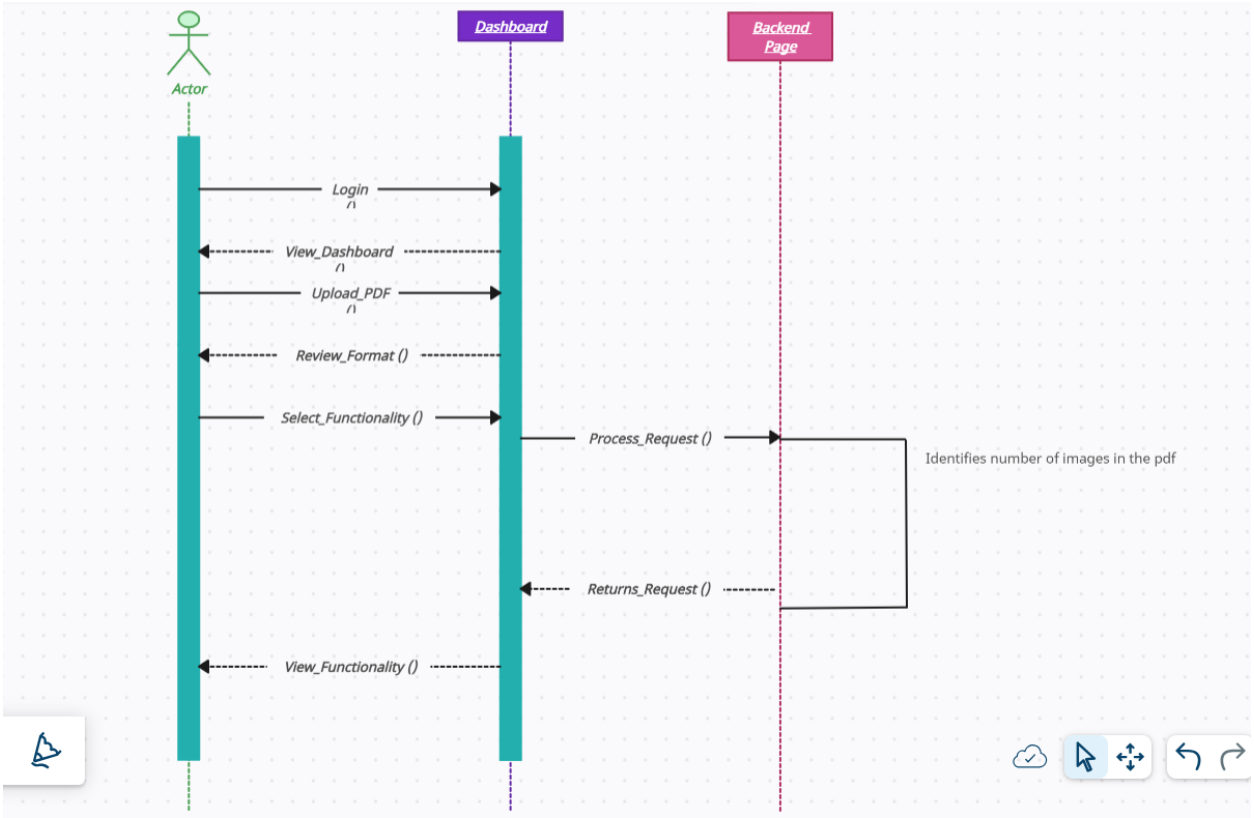
3.Upload PDF:


Actor
Dashboard
Authenticate
Login ()
View_Dashboard ()
Upload_PDF ()
Check_Format ()
Return_Format ()
Review_PDF_Format ()

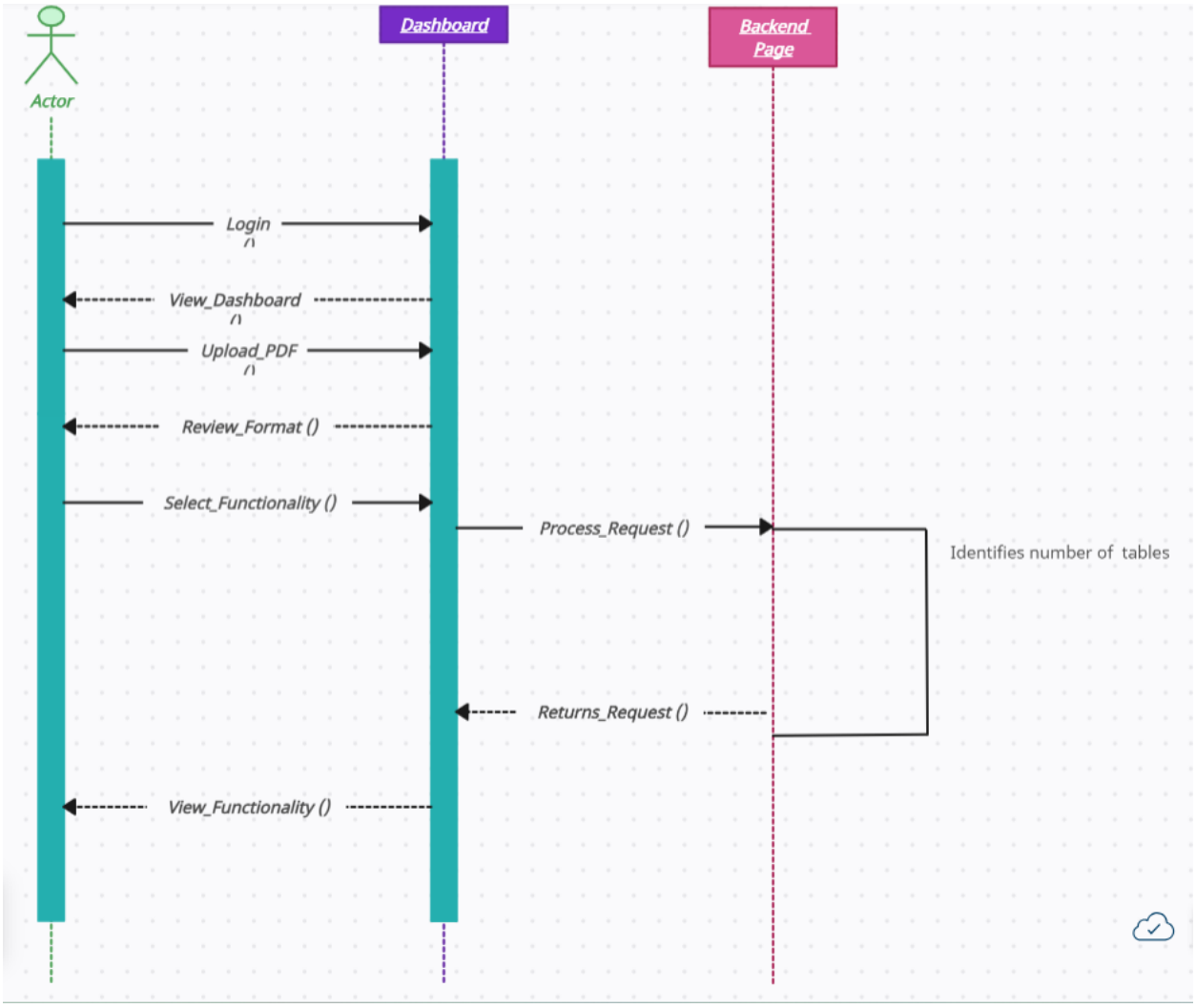
4. Font Evaluation :



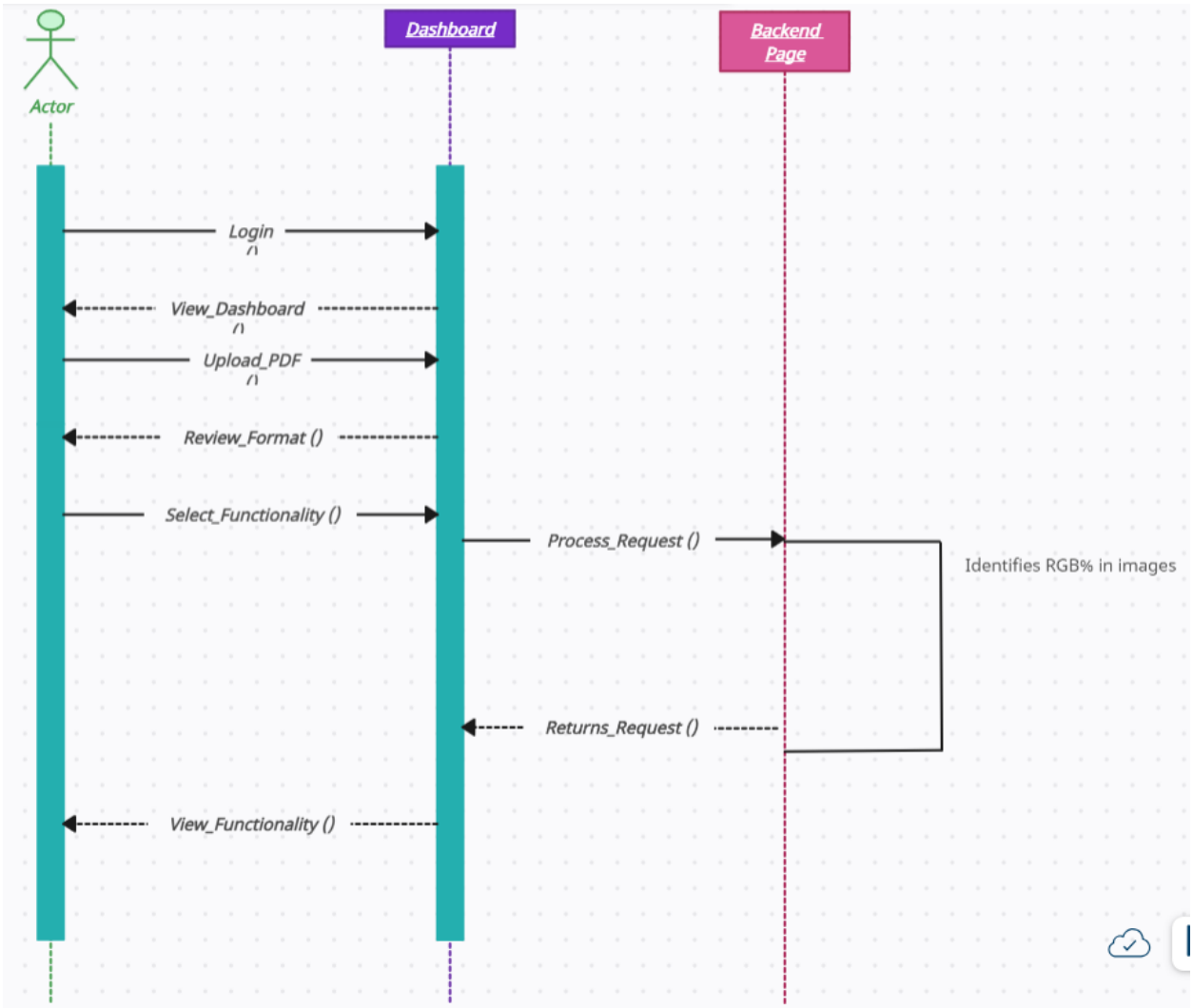
5. Identifying number of images:



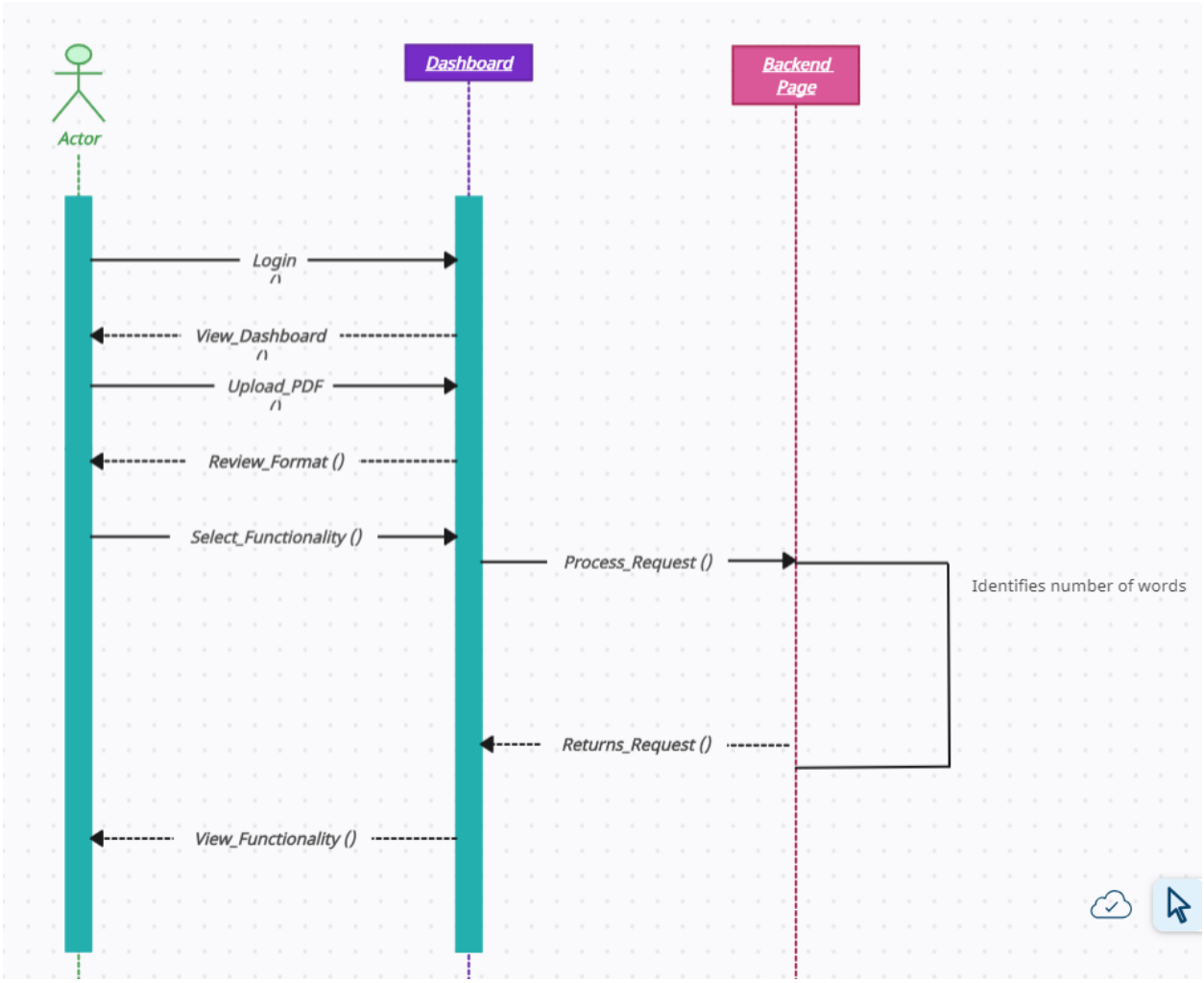
6. Identifies number of Tables:



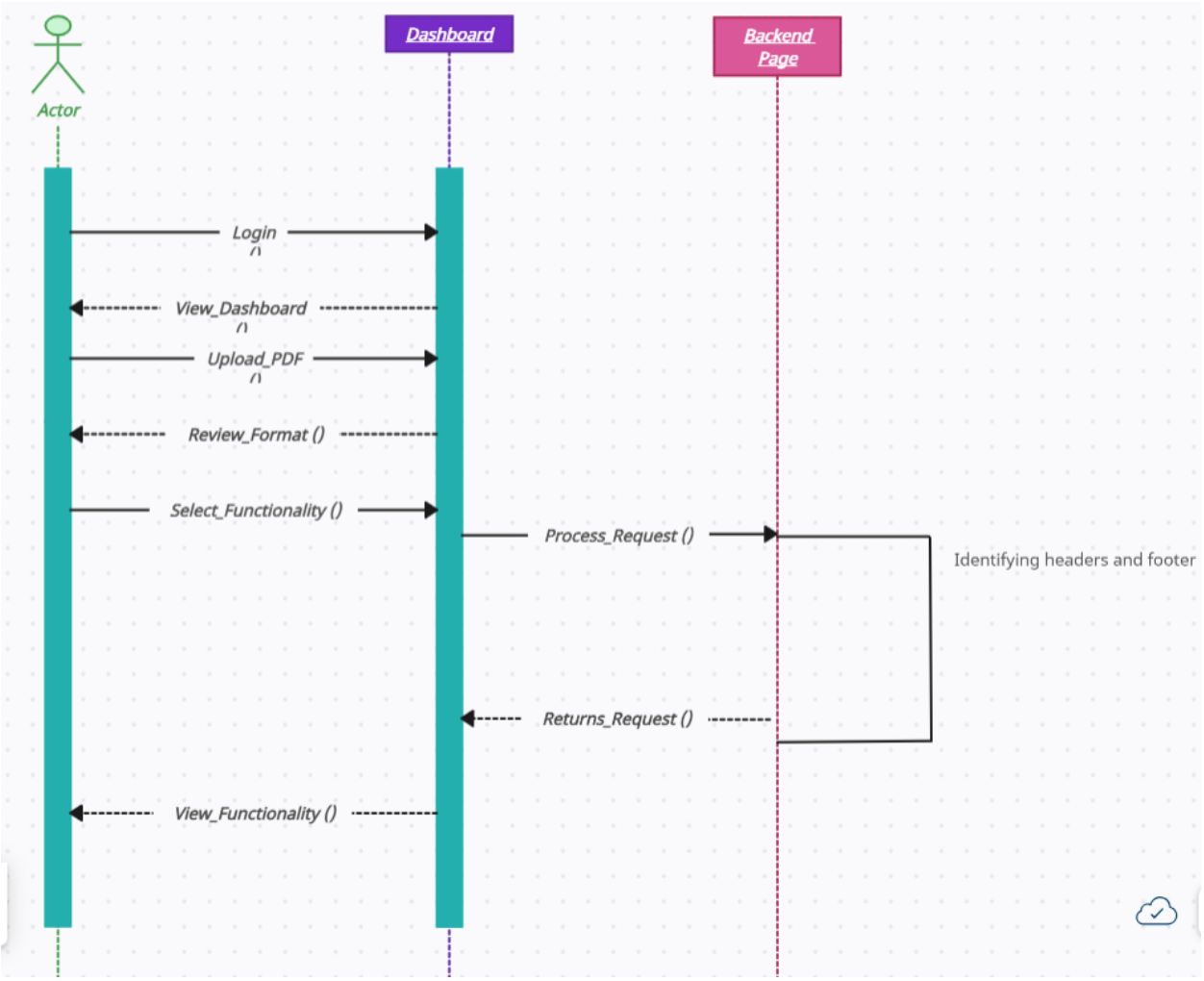
7. Identifies RGB% in images



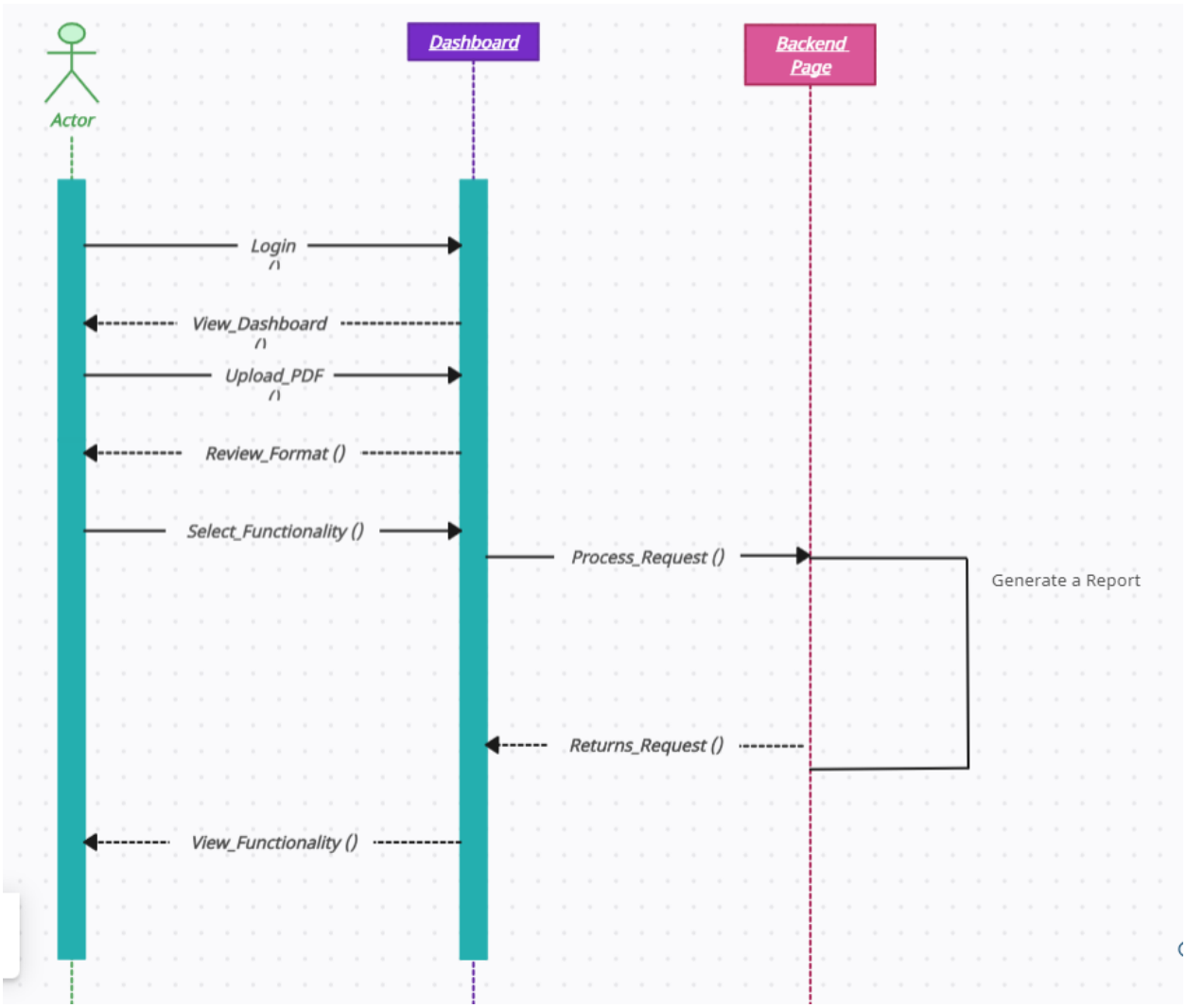
8. Identify number of Words:



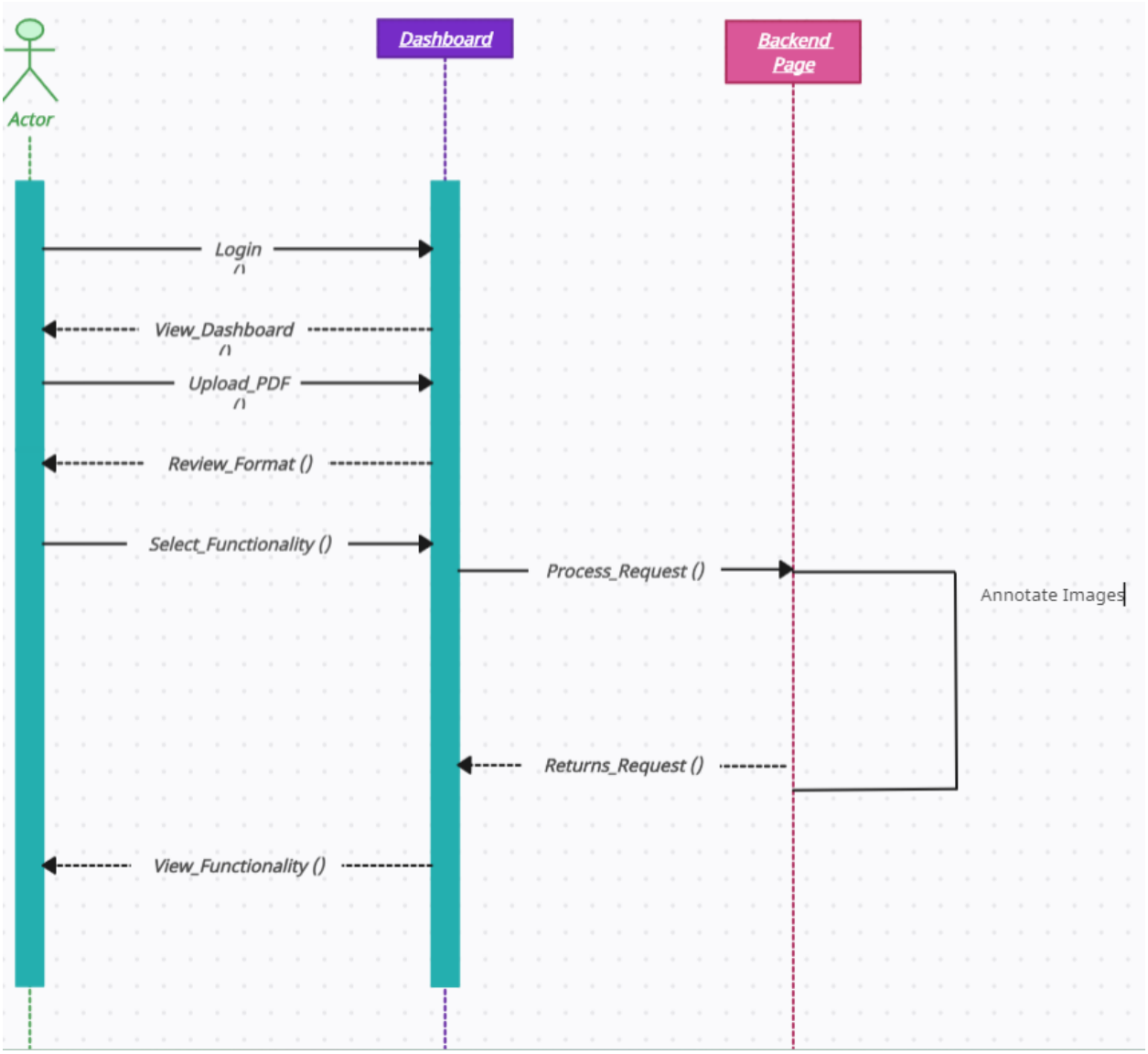
9. Identifying Headers and footers:



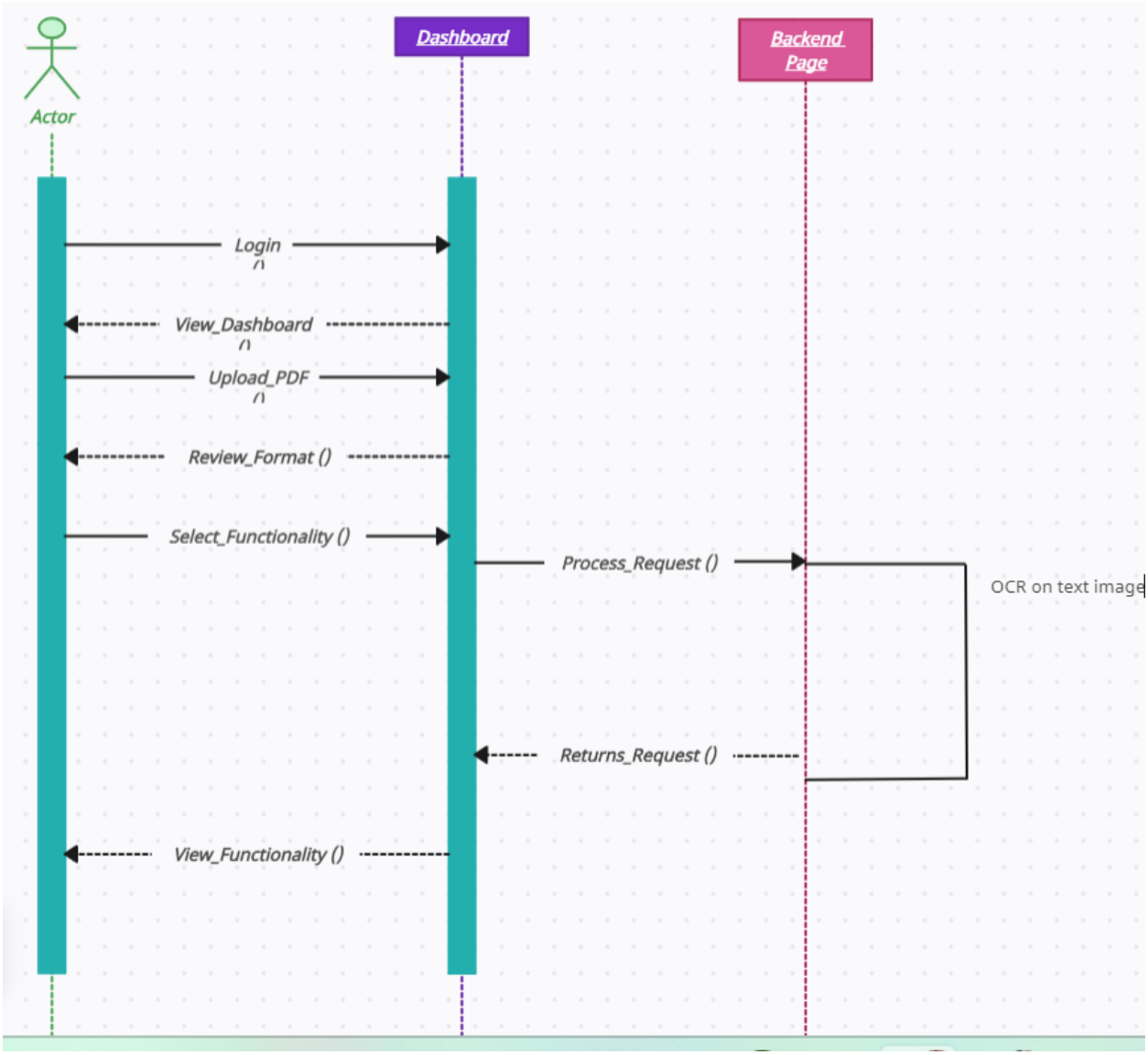
10. Generate a Report



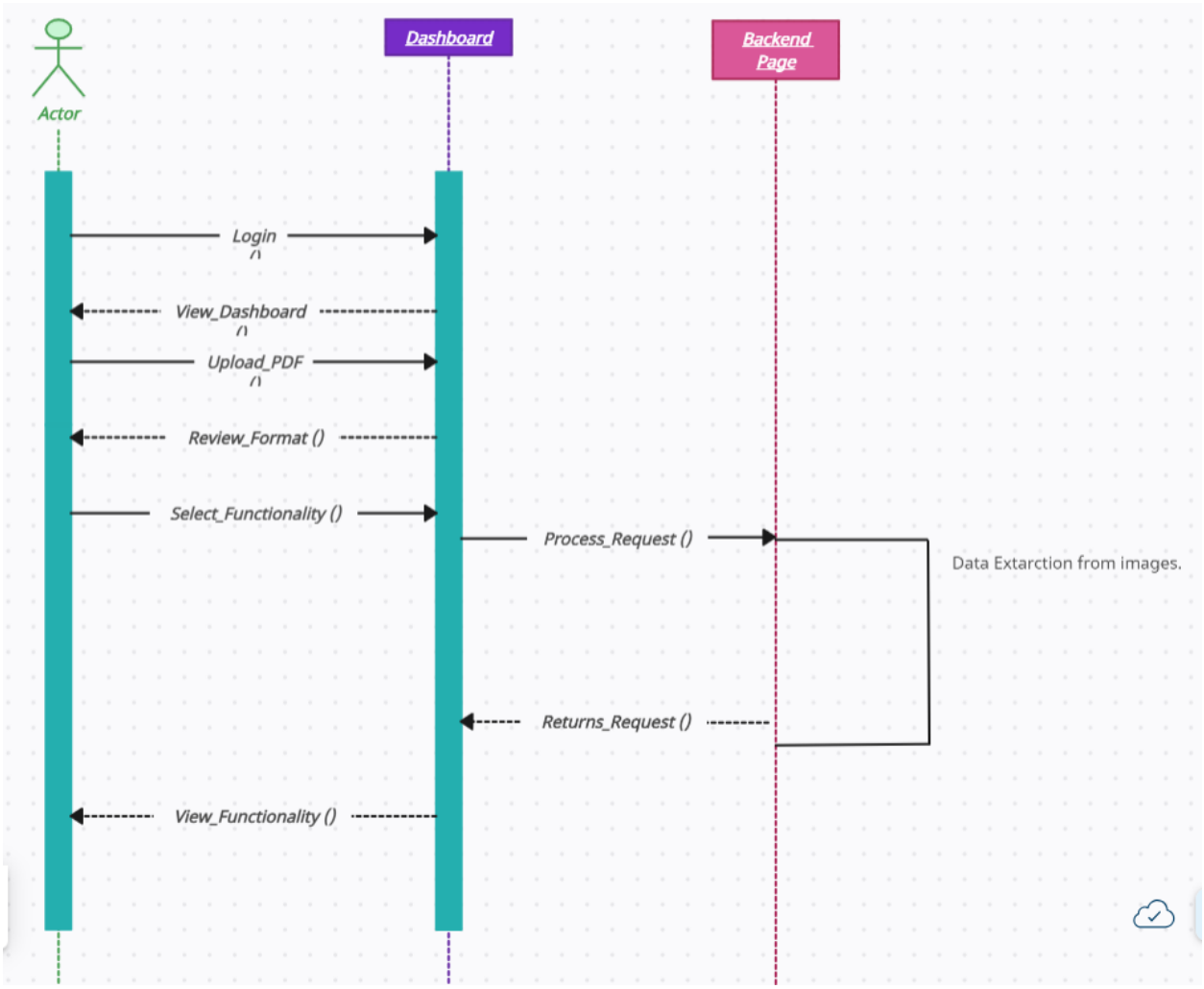
11. Annotate Images



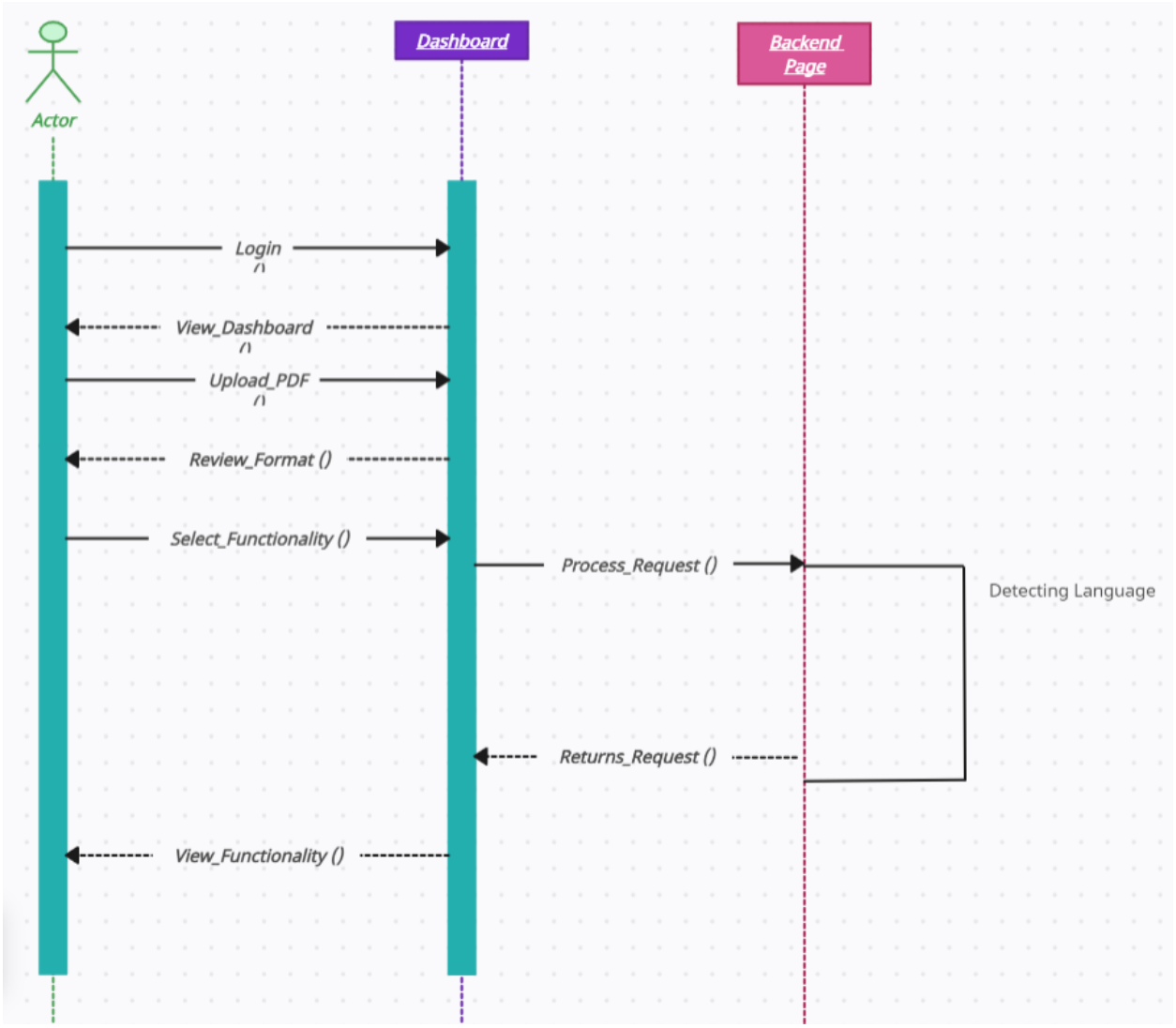
12. OCR On Text Image



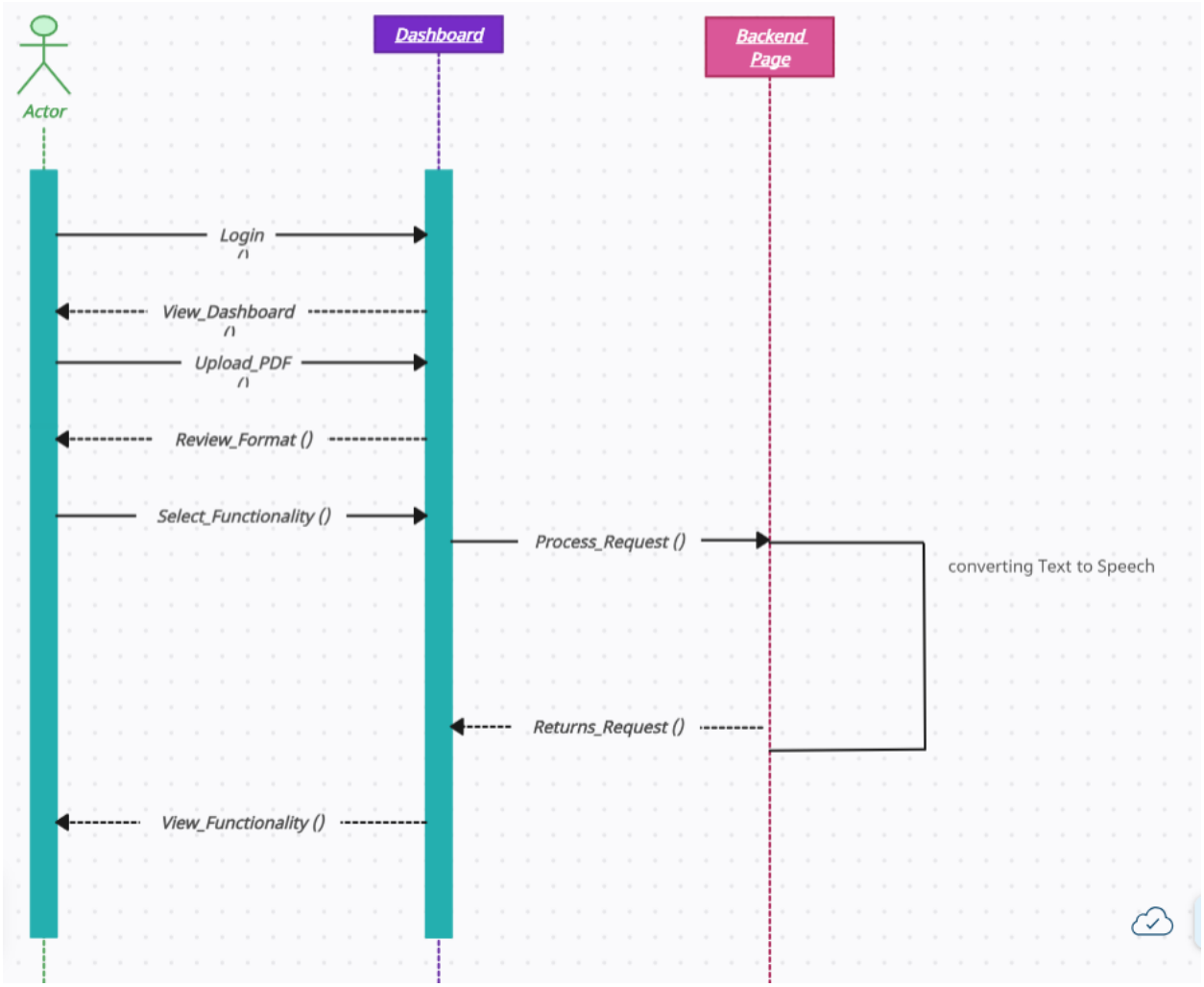
13. Data Extraction From Tables



14. Detecting language



15.Text to speech



Design Rationale

Reasons for Creating a PDF Evaluator

The requirements and goals of evaluating PDF files for accessibility serve as the foundation for the design rationale for the PDF evaluator system. Here are some significant design choices and their justifications:

Separating our concerns

Different classes are given responsibility for particular tasks under the system's design, which adheres to the idea of separation of concerns. Scalability and modularity are preserved as a result. For instance, the 'PDF File' class is in charge of managing file-related tasks, whereas the 'Evaluate Accessibility' class is responsible for assessing accessibility standards.

A user-centric strategy

The system's design is user-centric, and the 'User' class contains all user-related functionalities. Users are able to engage with PDF files and accessibility evaluations by registering, logging in, and doing so.

The system offers a straightforward and understandable user experience by segregating user-related tasks.

The architecture encourages modularity and extensibility, making it simple to incorporate new capabilities and standards. For instance, the functionality and properties associated with accessibility criteria are encapsulated in the Accessibility Criteria class. This enables simple criterion addition or adjustment without affecting other system components.

Encapsulation and Abstraction: To conceal implementation details and offer a clear interface for interacting with various components, the design places a strong emphasis on encapsulation and abstraction. Each class isolates its attributes and associated capabilities, improving code organization and maintainability.

Report Generation:

The framework incorporates an Accessibility Report class to create reports in light of the assessment results. This class epitomizes the report-related qualities and functionalities, giving an organized portrayal of the assessment result. Producing reports helps in imparting the assessment results to the users, empowering them to make fitting moves.

Flexibility in File Processing:

The PDF Document class incorporates techniques for file upload, validation, and processing. This considers adaptability in taking care of various sorts of PDF documents and performing vital tasks to guarantee information trustworthiness and precision during the assessment cycle.

Collaboration and Communication:

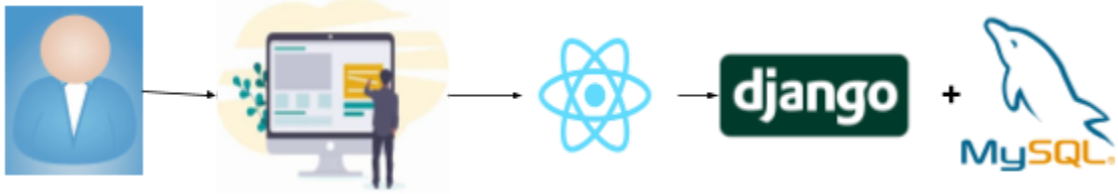
The plan works with coordinated effort and correspondence between various parts. For instance, the evaluate accessibility class connects with the PDF Document class to get to the PDF content and with the accessibility criteria class to recover the models for assessment. This joint effort guarantees that the assessment cycle is directed precisely and effectively.

User-Friendly Interfaces:

The plan expects to give easy to use connection points to clients to cooperate with the framework. This incorporates functionalities like enrollment, login, and intuitive steps for transferring PDFs, choosing assessment measures, and reviewing reports. Easy to understand interfaces improve the convenience and overall client experience of the framework.

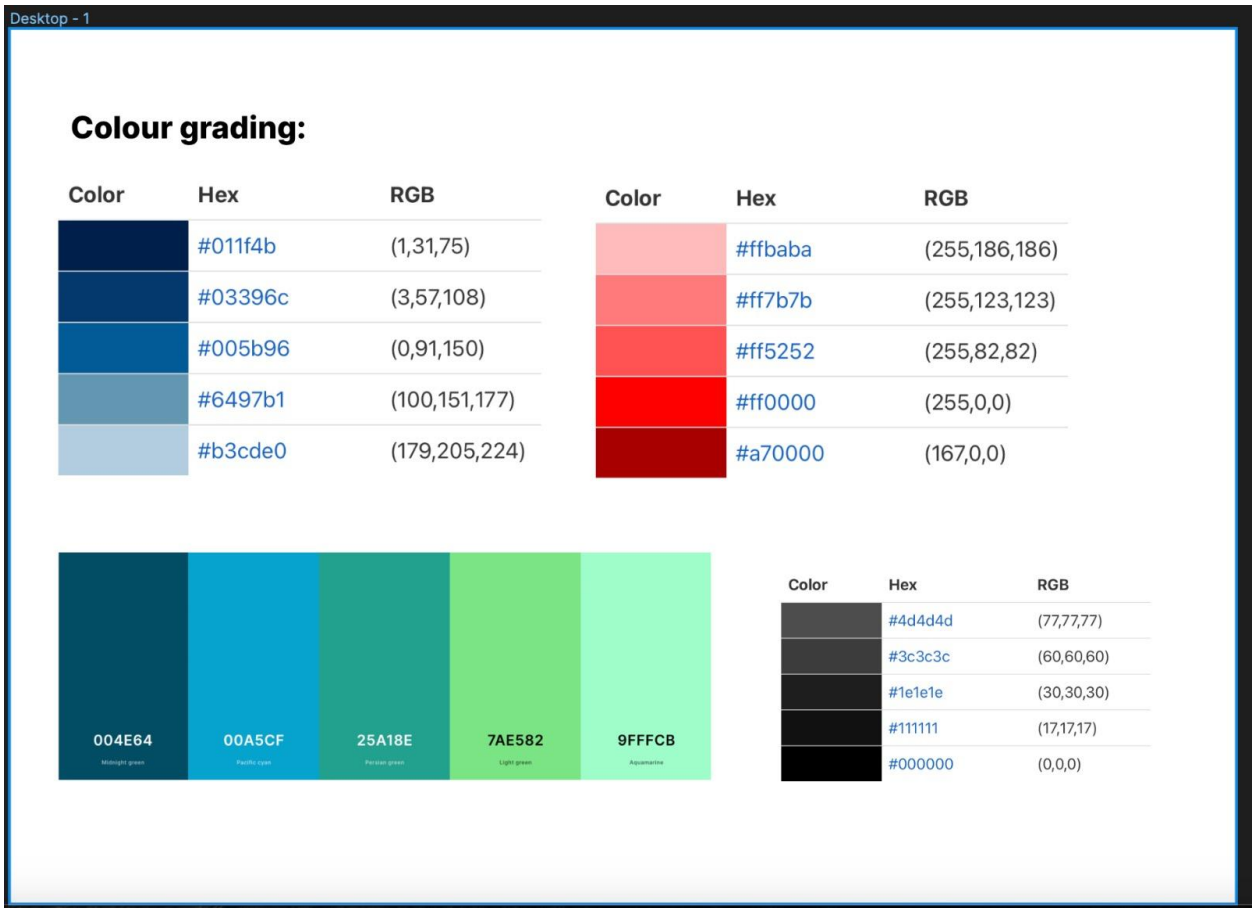
Overall, the design rationale for the PDF evaluator framework centers around particularity, adaptability, client centricity, and successful correspondence. These plan choices work with proficient assessment of PDF records for openness, enabling clients to survey and work on the availability of their archives.

Information Architecture Diagram



User Interface Wireframe(s)/Screenshot(s)

- Color Grading



- Stylesheet
- We are going to use Bootstrap to style the UI.

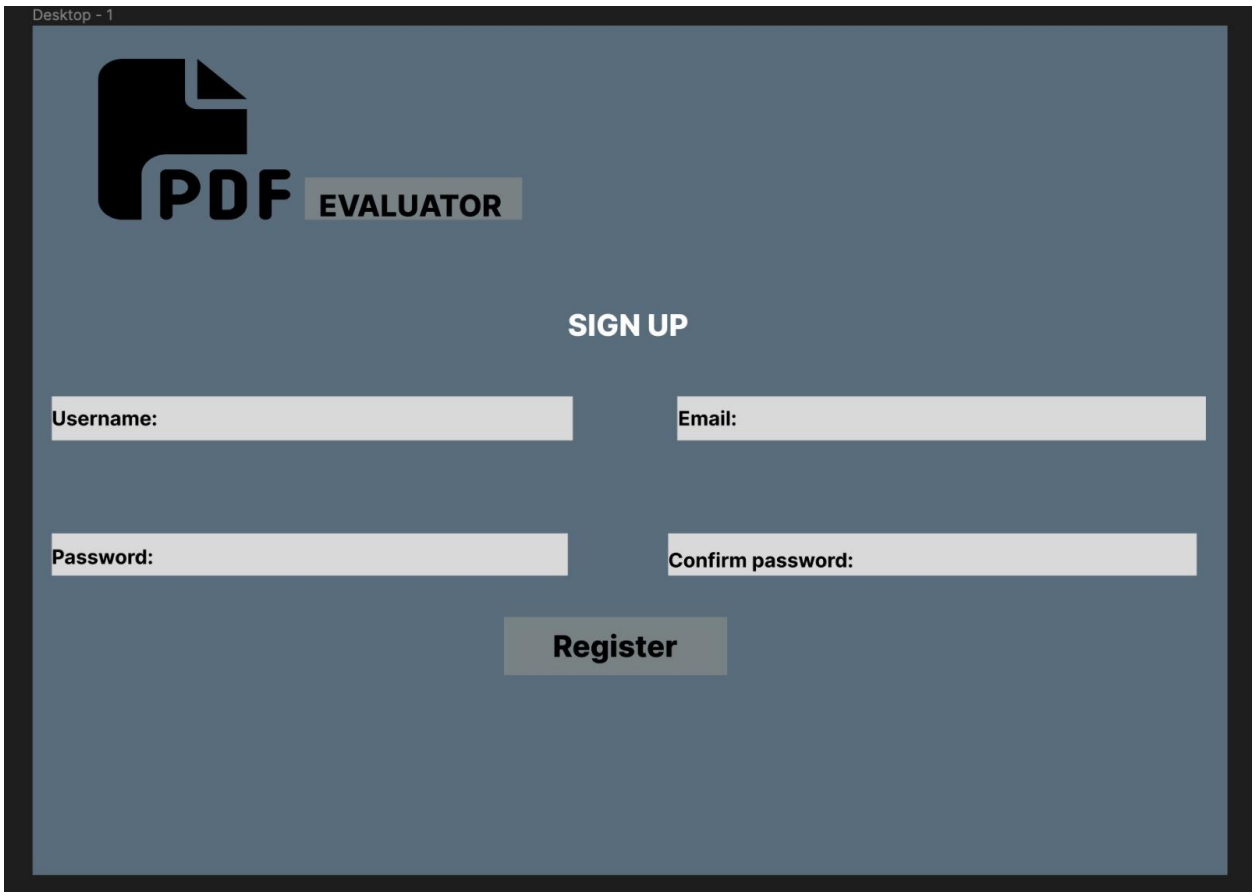
- Logo

1. Welcome Page :

- Here we are displaying the welcome page of our pdf evaluation project.

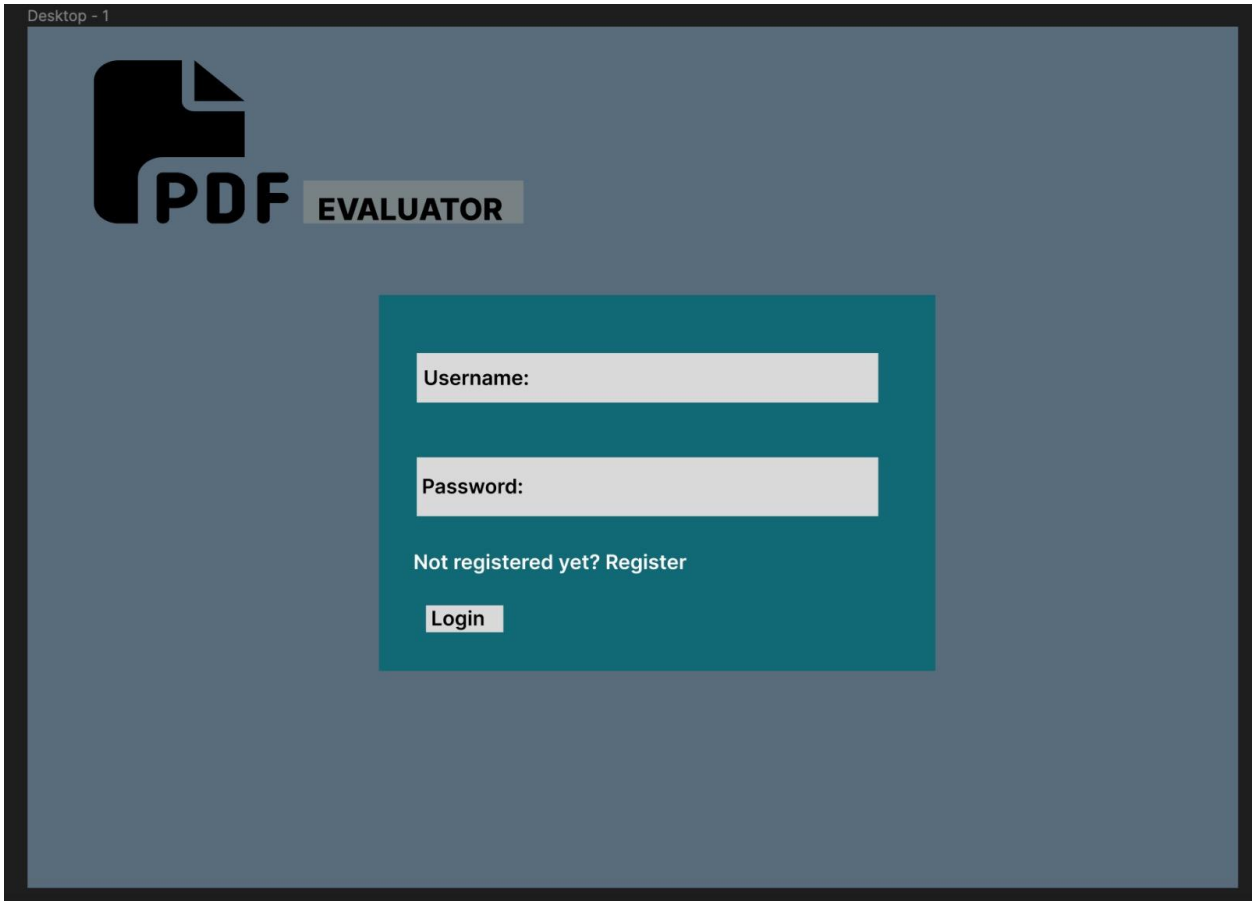
2. Registration Page :

- This is the registration page of our project which shows the registration process to move further.


Desktop - 1
PDF
EVALUATOR
SIGN UP
Username:
Email:
Password:
Confirm password:
Register

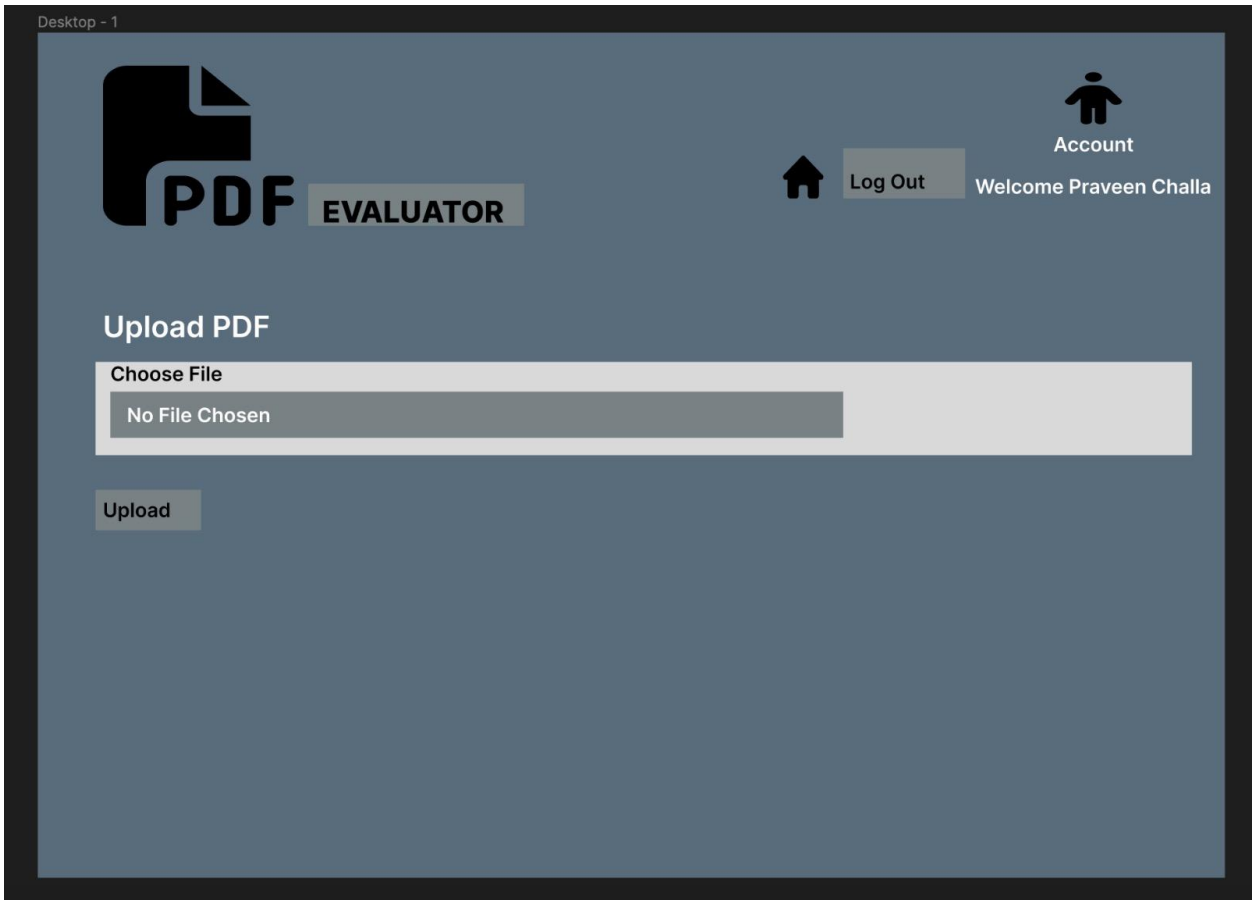
3. Login Page :

- This shows the login portal of our project in order to enter the website and check out.



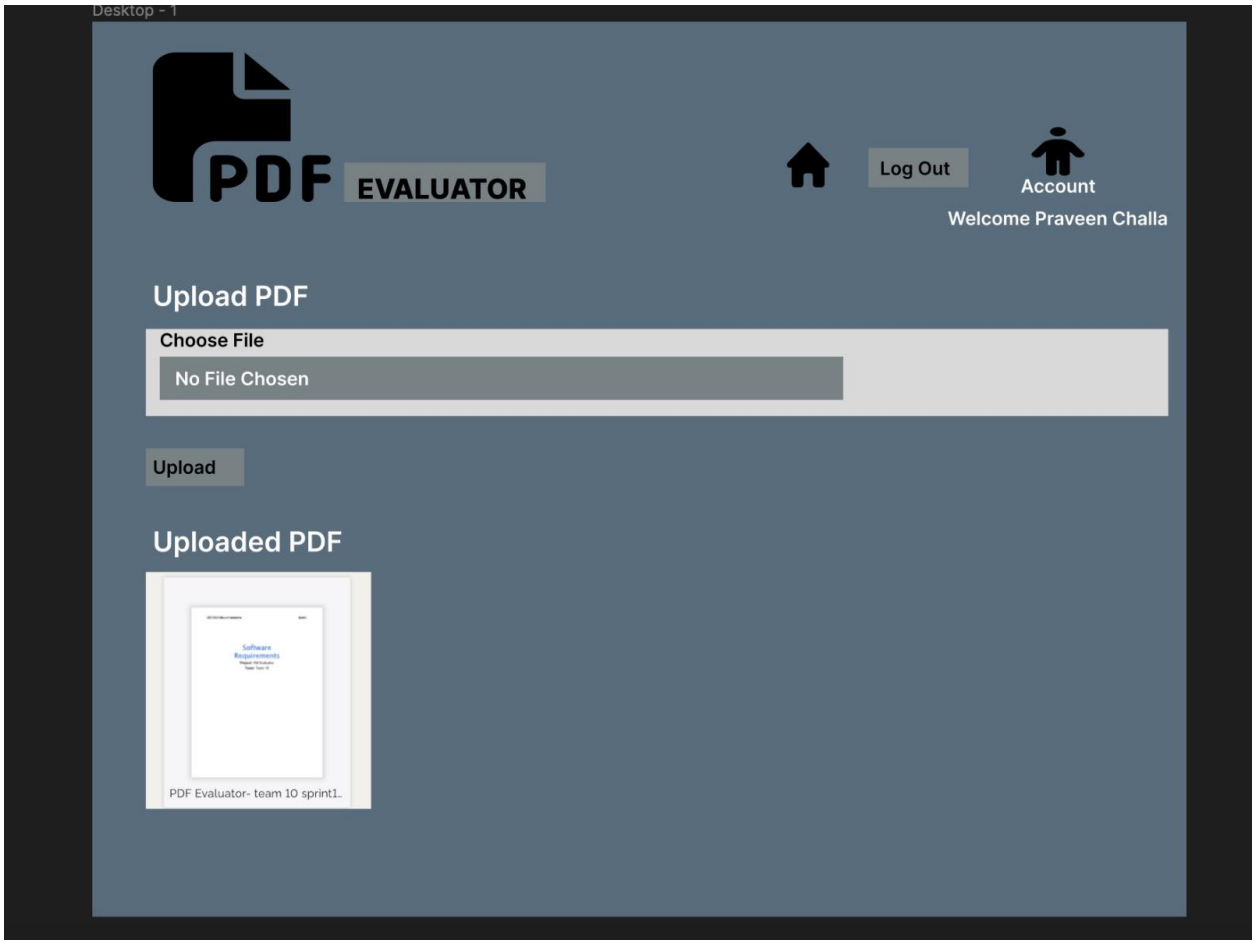
4. Dashboard :

- Here we can view the dashboard page of our portal which shows different kinds of activities that can be performed for the results.
- In this page, we can upload any kind of pdf for evaluating the features in it.



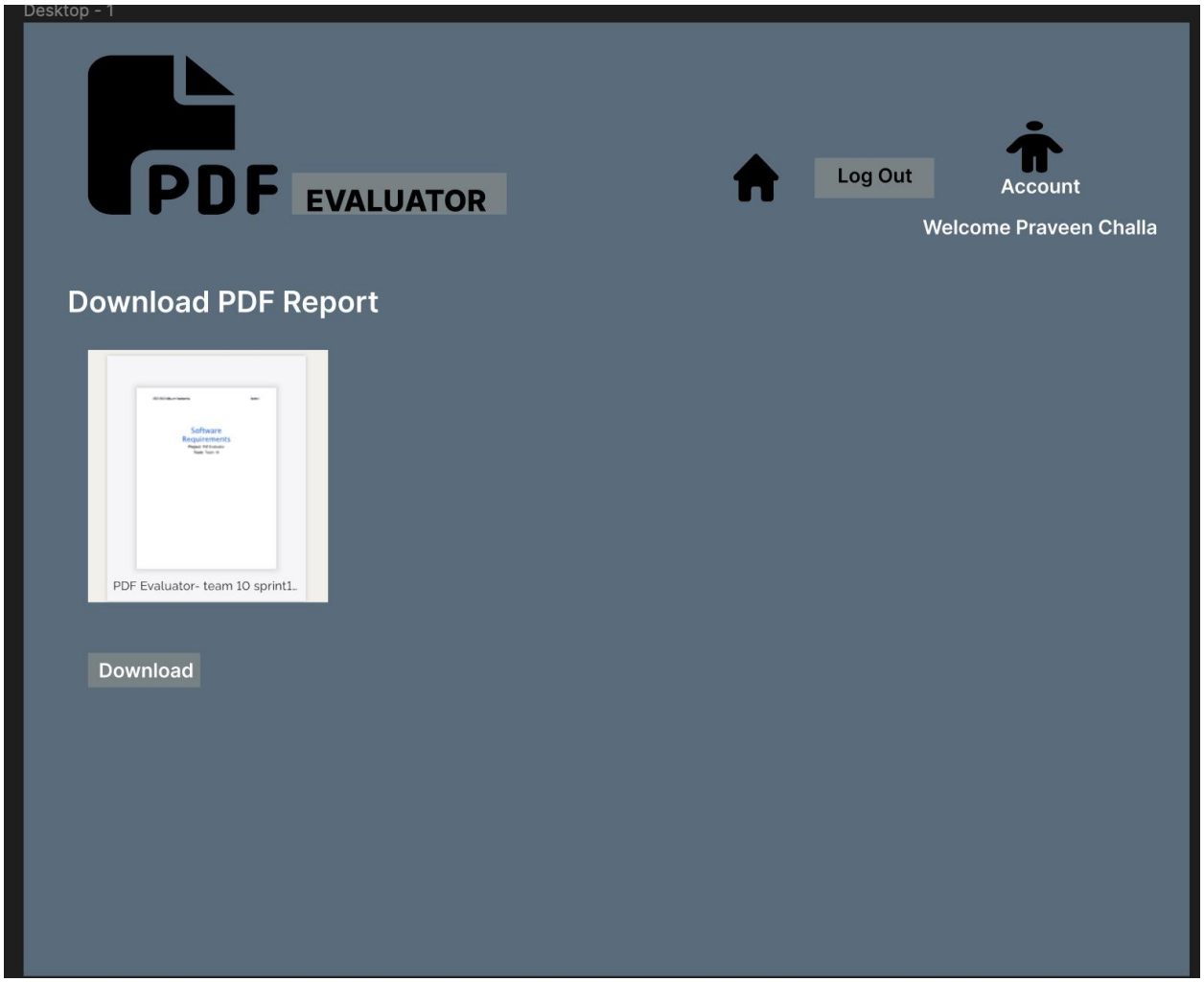
5. PDF Visualizer Page :

- Here we can see the pdf file uploaded as a preview (which is basically the first page of the document).



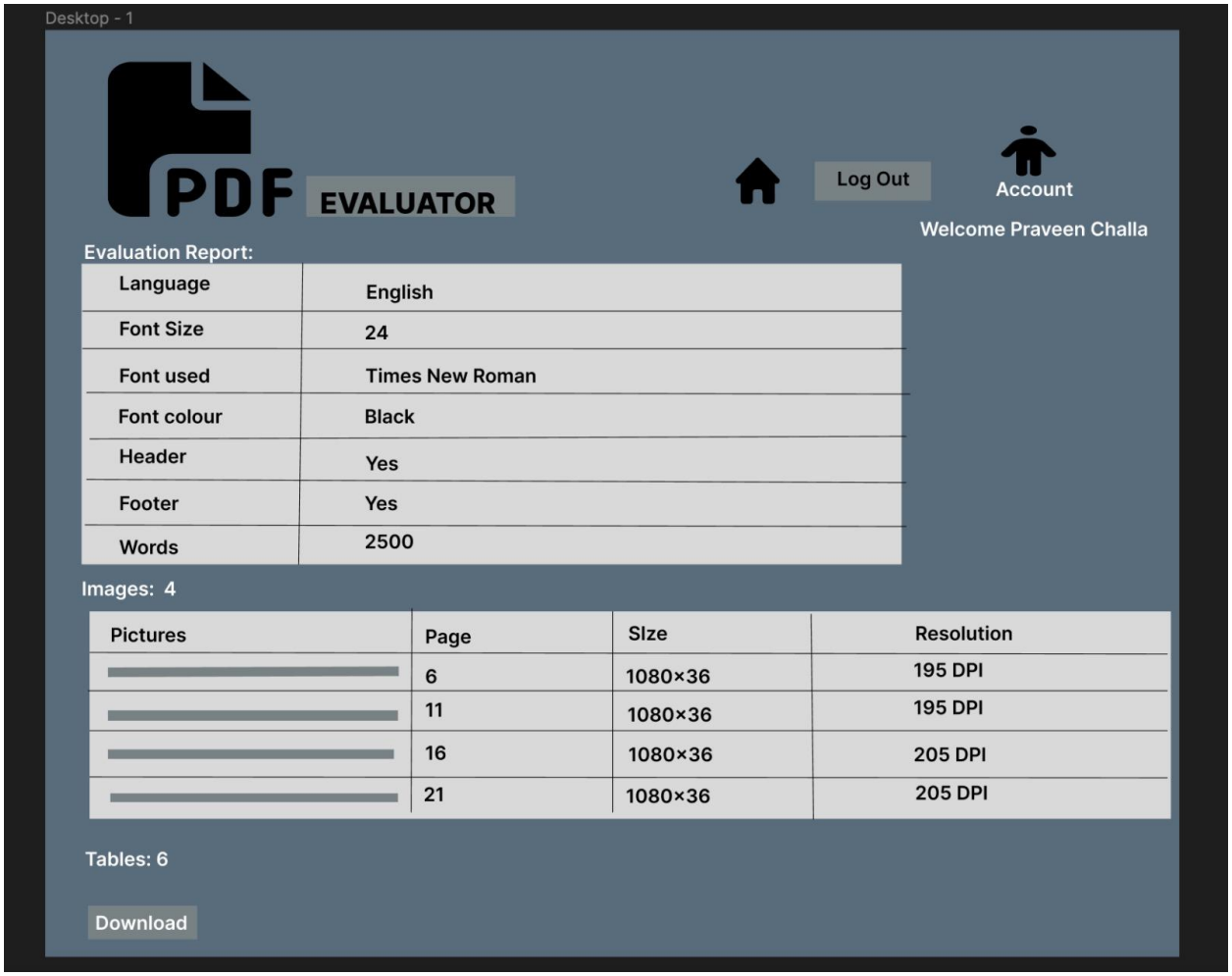
6. PDF Download Page :

- In this page, we can download the pdf file uploaded for the reference if you feel that any wrong file is uploaded.



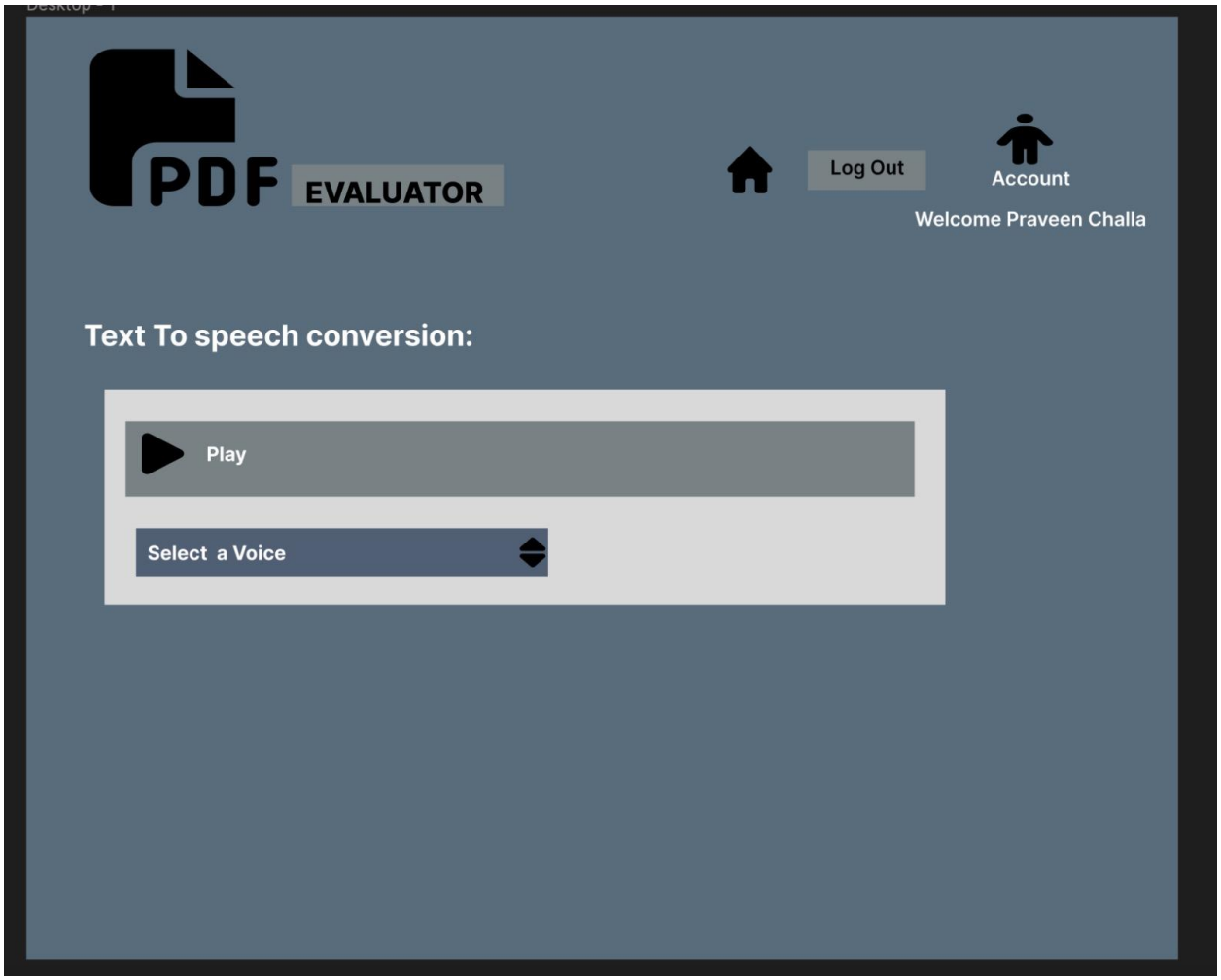
7. Features Evaluation Page :

- This is the most important functionality page of our project. Here we can evaluate the pdf file and see different types of results like the language of the text in the pdf file, font size,color and type of the text, if there are headers and footers , number of images and their corresponding page numbers and number of tables.



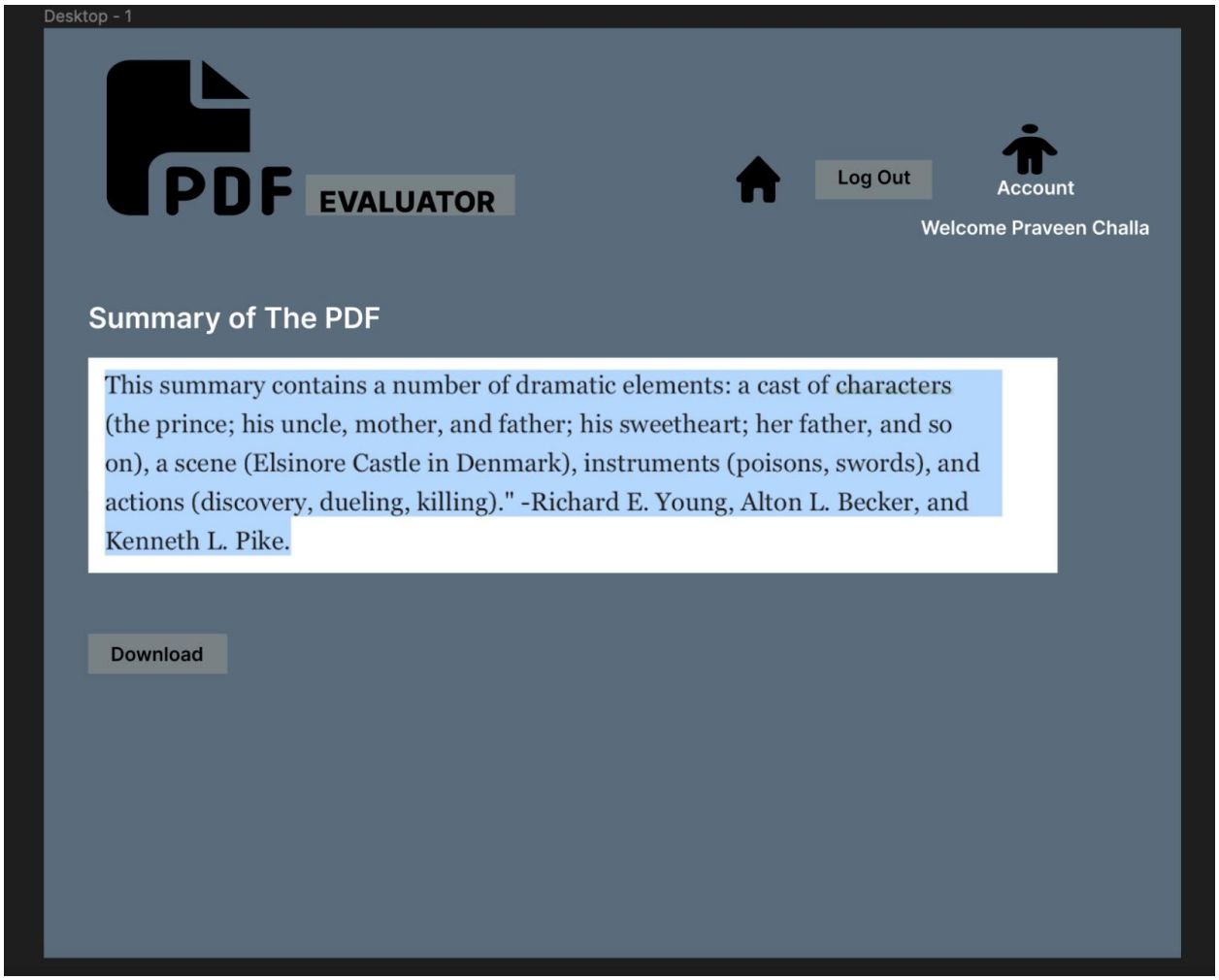
8. Text to Speech Conversion Page :

- This page is generally used to perform text to speech operation which is nothing but there is a mike option which on clicking will read the text in the pdf file.
- We can simply click play button to generate speech functionality of the pdf text.



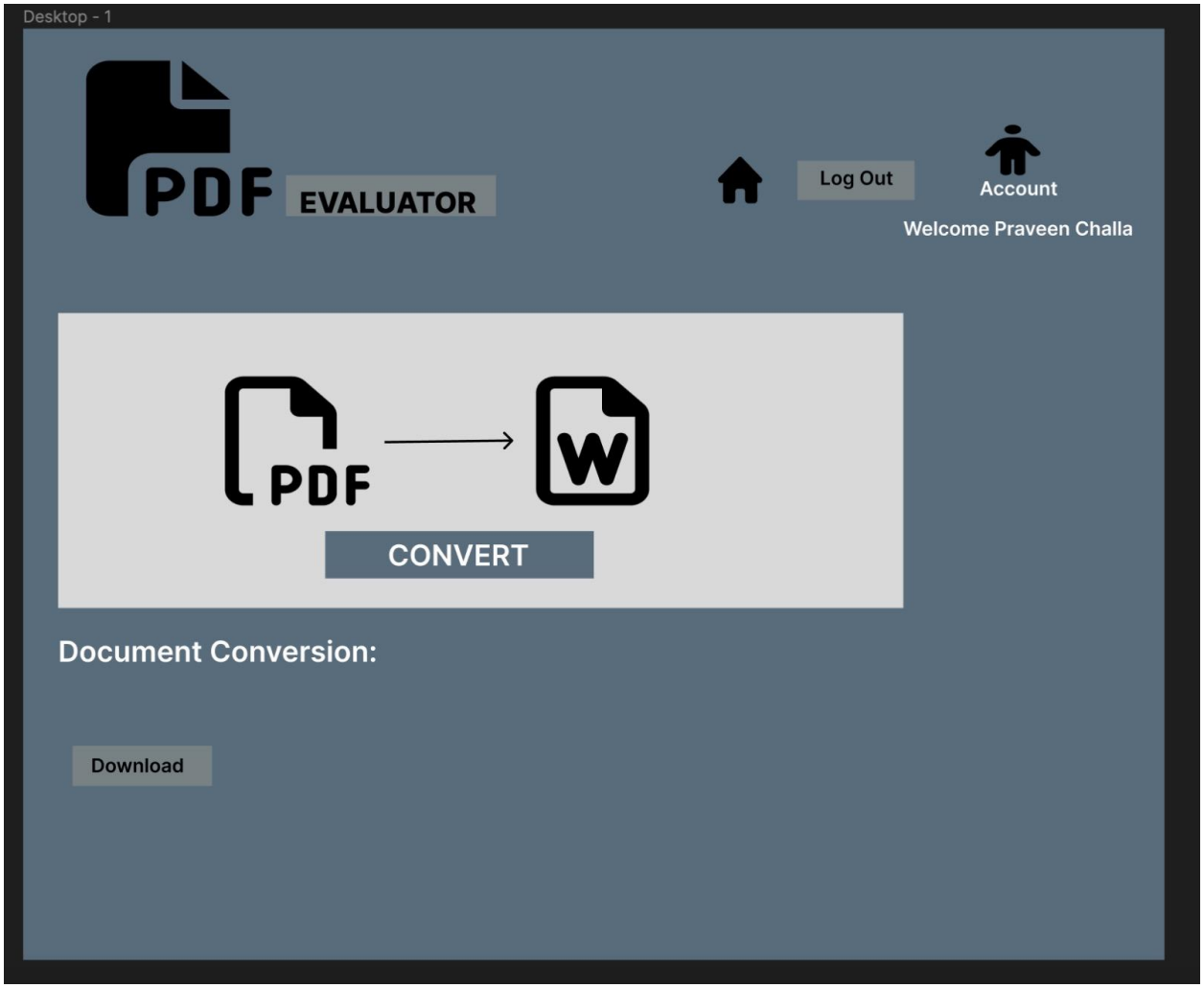
9. Summary Page :

- In this page, we will get the summary or a report kind of the information available in the file based on the important concepts which helps us to get the abstract idea of the pdf file uploaded.



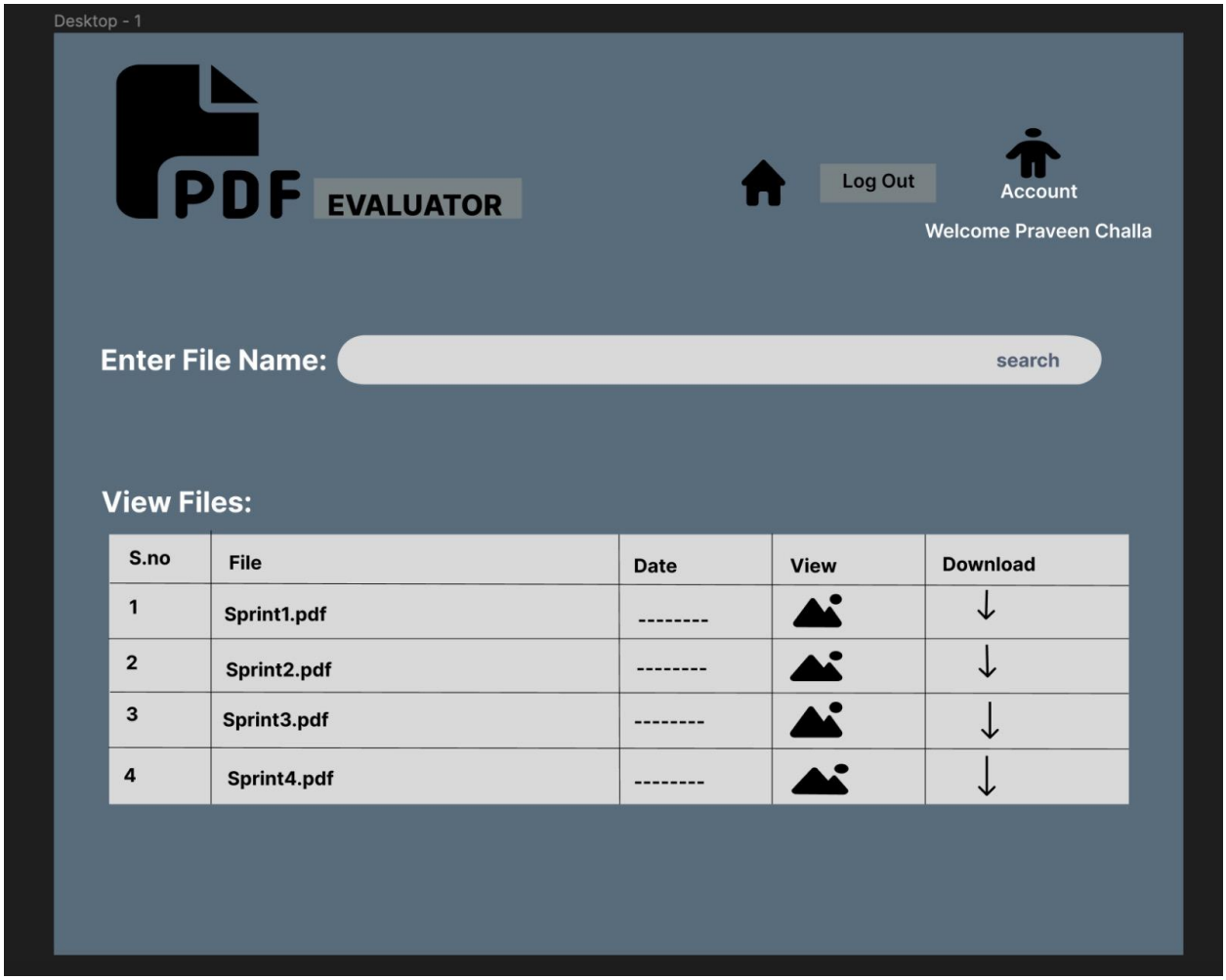
10. PDF Converter Page :

- In this page, there is an option for converting the uploaded pdf file into word format which helps us to perform editing functionality to make necessary changes based on the needs.
- We can convert the file into other first and simply click download button to download the file in the converted format.



11. PDF Search Page :

- This page allows us to search for the required pdf file by simply searching in the search bar with the corresponding name of the file we want.
- Here a list of pdf files are shown and we can preview the pdf we want and after searching the required pdf we can download the file.



Individual Contribution:

S.No	Name	Student Id	Contribution
1	Chandrika Vasantha	11608413	Worked on Sequence Diagrams, Information Architecture diagram
2	Tarun Teja Pasupuleti	11601673	Worked on Class Diagram, Sequence Diagram, Design rationale, user interface.
3	Samanthula Aasritha Reddy	11654299	Worked on ER Diagram, Sequence Diagram, Design Rationale, Document.
4	Jaidithya Yarava	11646470	Worked on Class Diagram, Sequence Diagram, ER diagram.
5	Praveen Challa	11604608	Worked on Sequence Diagram, User interface, Document.