|  |  |  |
| --- | --- | --- |
| **Inżynieria programowania – Projekt** | | |
| Grupa: 3ID14B | Zespół:  Adrian Kowalski  Paweł Czyżewski  Yevhenii Mukha | Data: 03.02.2022 |
| Temat: Aplikacja do zarządzania dokumentacją firmy | | |

**Opis systemu**

Jest to aplikacja do zarządzania dokumentacją firmy. Dzięki niej możemy prowadzić swobodną wymianę plików bez potrzeby dostarczania fizycznych dokumentów.

Przy jej użyciu pracownik może dostarczać wnioski i podania do kierownika, które następnie, jeżeli zostaną zatwierdzone trafiają do kadr, a potem do archiwum.

Aplikacja posiada różne opcje w zależności od tego na jakim koncie obecnie użytkownik jest zalogowany.

Pracownicy mogą składać wnioski, oraz wrzucać do swojego katalogu pliki niezbędne do ich pracy w firmie.

Kierownik może umieszczać faktury, oraz historie transakcji w swoim katalogu, a także zatwierdzać wnioski pracowników, a następnie przesyłać je do kadr w celu dalszego zweryfikowania podań.

Kadry mogą przechowywać w swoim katalogu między innymi listy pracowników, a także zatwierdzać wnioski przesłane przez kierownika do zweryfikowania. Wnioski te zostają następnie przez kadry wysłane do katalogu archiwum, gdzie są magazynowane.

**Sposób instalacji**

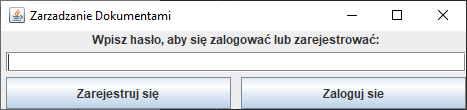
W celu uruchomienia aplikacji wymagane jest posiadanie oprogramowania Java (preferowana najnowsza wersja) (<https://java.com/pl/download/>). Aby dokonać uruchomienia należy kliknąć na aplikację (rozszerzenie .jar) prawym przyciskiem myszy, a następnie wybrać opcję otwórz za pomocą i wybrać z listy **Platforma Java (TM) SE binarna.**

**Zastosowane algorytmy**

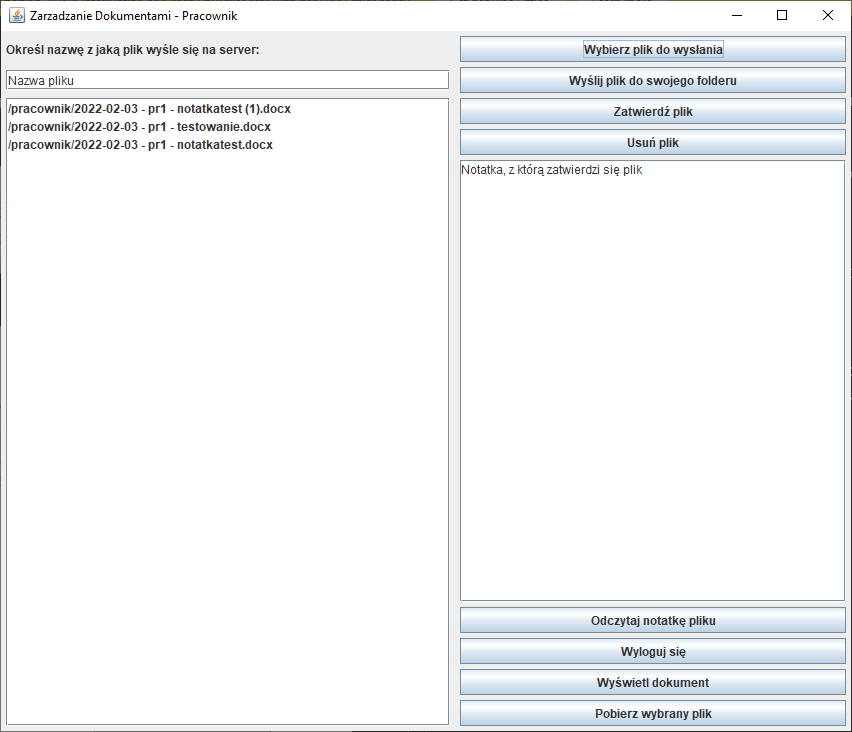
**W naszym programie została użyta biblioteka Swing w celu stworzenia interfejsu użytkownika. Wykorzystaliśmy również API serwera zewnętrznego, aby przechowywać dokumentację firmy. Uważamy, że obecne praktyki utrzymywania danych w chmurze są popularne i bezpieczne w porównaniu do tradycyjnych nośników fizycznych posiadanych przez pracowników czy pracodawców w firmach.**

**Obsługa program**

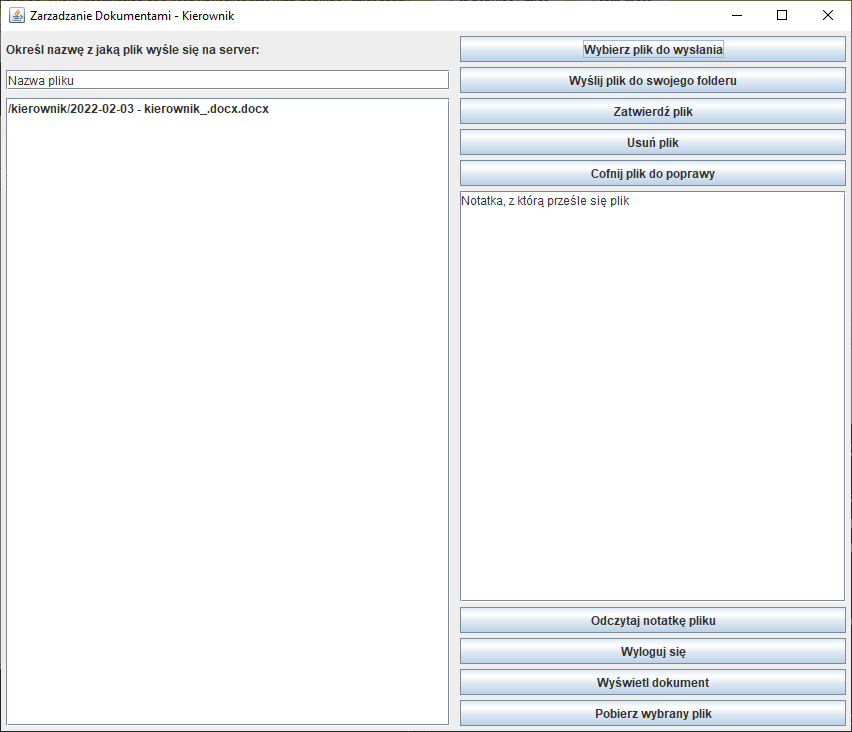
**Ekran logowania**



**Interfejs pracownika**

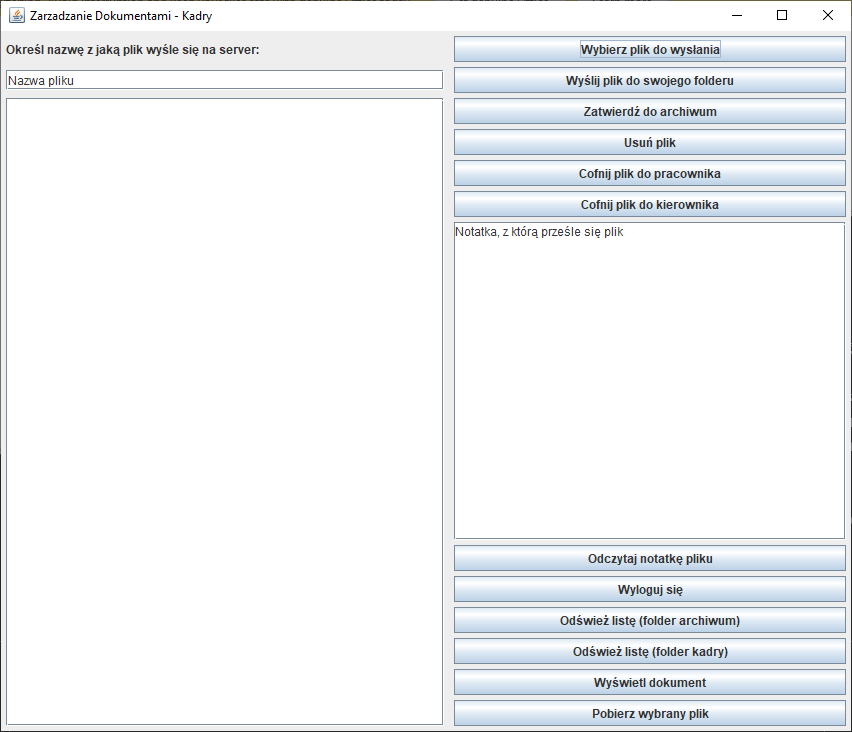


**Interfejs kierownika**



1. Wybierz plik do wysłania – jest to guzik dzięki któremu możemy wybrać plik który chcemy wysłać do Kadr.
2. Wyślij plik do swojego folderu – jest to guzik odpowiedzialny za umieszczanie wybranego pliku w swoim katalogu.
3. Zatwierdź plik – dzięki temu możemy przesłać dalej wniosek otrzymany przez pracownika.
4. Usuń plik – usuwa plik z katalogu.
5. Cofnij plik do poprawy – służy do odesłania wniosku do pracownika w celu jego poprawy.
6. Odśwież listę plików – przeładowuje on listę plików, tak aby była możliwość zobaczenia aktualnej listy.
7. Pobierz wybrany plik – po naciśnięciu możemy pobrać plik ze swojego katalogu.
8. Lista plików – tutaj wyświetlane są wszystkie pliki znajdujące się obecnie w katalogu.
9. Nazwa pliku – to okno jest ściśle powiązane z przyciskiem ‘Wybierz plik do wysłania’, ponieważ to tu wyświetlana jest nazwa pliku który chcemy wysłać.

**Interfejs kadr**



**Klasy i metody:**

Klasa Main – główna klasa programu, zawarte są w niej między innymi metody najczęściej używane i wywoływane na przyciskach.

Klasa DocumentPreview – klasa służąca do wyświetlania dokumentu w GUI

Klasa PasswordHash – klasa służąca do hashowania i walidowania haseł użytkowników.

Klasa Pracownik – klasa zawierająca interfejs pracownika (ekrany, przyciski).

Klasa Kierownik – klasa zawierająca interfejs kierownika (ekrany, przyciski).

Klasa Kadry – klasa zawierająca interfejs Kadr (ekrany, przyciski).

Metody klasy Main:

(package private) static void

[**centreWindow**](file:///C:\Users\kowal\Desktop\ProjektIPJAVA\doc\com\company\Main.html#centreWindow(java.awt.Window))([**Window**](https://docs.oracle.com/en/java/javase/17/docs/api/java.desktop/java/awt/Window.html) frame)

Metoda służy do zmiany pozycji GUI na ekranie.

(package private) static void

[**chooseFile**](file:///C:\Users\kowal\Desktop\ProjektIPJAVA\doc\com\company\Main.html#chooseFile())()

Metoda służy do wybrania plików z komputera użytkownika.

(package private) static void

[**deleteFile**](file:///C:\Users\kowal\Desktop\ProjektIPJAVA\doc\com\company\Main.html#deleteFile(java.lang.String))([**String**](https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/String.html) pathDeleteDestination)

Metoda służy do usunięcia wybranego pliku z serwera.

(package private) static void

[**displayGUIComponents**](file:///C:\Users\kowal\Desktop\ProjektIPJAVA\doc\com\company\Main.html#displayGUIComponents())()

Metoda głównie służy do wyświetlenia interfejsu.

(package private) static void

[**downloadFile**](file:///C:\Users\kowal\Desktop\ProjektIPJAVA\doc\com\company\Main.html#downloadFile(java.lang.String,java.io.File))([**String**](https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/String.html) sourcePath, [**File**](https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/io/File.html) destinationPath)

Metoda do zgrywania plików z serwera.

(package private) static void

[**generateFilePreview**](file:///C:\Users\kowal\Desktop\ProjektIPJAVA\doc\com\company\Main.html#generateFilePreview(java.lang.String))([**String**](https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/String.html) sourcePath)

Metoda pozwalająca wyswietlic dokument

(package private) static void

[**generateNote**](file:///C:\Users\kowal\Desktop\ProjektIPJAVA\doc\com\company\Main.html#generateNote(java.lang.String))([**String**](https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/String.html) noteDestination)

Metoda pozwalająca wygenerować notatkę do pliku

static void

[**infoBox**](file:///C:\Users\kowal\Desktop\ProjektIPJAVA\doc\com\company\Main.html#infoBox(java.lang.String,java.lang.String))([**String**](https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/String.html) infoMessage, [**String**](https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/String.html) titleBar)

Wyświetla wiadomość zwrotną do użytkownika

(package private) static void

[**listFiles**](file:///C:\Users\kowal\Desktop\ProjektIPJAVA\doc\com\company\Main.html#listFiles(java.lang.String))([**String**](https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/String.html) displayPath)

Metoda wyświetla listę wszystkich plików na serwerze.

(package private) static void

[**listFilesForPracownik**](file:///C:\Users\kowal\Desktop\ProjektIPJAVA\doc\com\company\Main.html#listFilesForPracownik())()

Metoda służy do wyświetlania zawartości plików na serwerze dla załogowanego pracownika.

(package private) void

[**logowanie**](file:///C:\Users\kowal\Desktop\ProjektIPJAVA\doc\com\company\Main.html#logowanie(char%5B%5D))(char[] password)

Metoda służy do logowania w aplikacji jako kierownik/kadry/pracownik.

static void

[**main**](file:///C:\Users\kowal\Desktop\ProjektIPJAVA\doc\com\company\Main.html#main(java.lang.String%5B%5D))([**String**](https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/String.html)[] args)

Metoda główna, która uruchamia się program.

(package private) static void

[**moveFile**](file:///C:\Users\kowal\Desktop\ProjektIPJAVA\doc\com\company\Main.html#moveFile(java.lang.String,java.lang.String))([**String**](https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/String.html) sourcePath, [**String**](https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/String.html) destinationPath)

Metoda służy do przesuwania plików na serwerze z określonego folderu do innego.

(package private) static void

[**readNote**](file:///C:\Users\kowal\Desktop\ProjektIPJAVA\doc\com\company\Main.html#readNote(java.lang.String))([**String**](https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/String.html) noteDestination)

Metoda pozwalająca odczytać notatkę dodaną do pliku przy jego transferze

(package private) static void

[**saveFile**](file:///C:\Users\kowal\Desktop\ProjektIPJAVA\doc\com\company\Main.html#saveFile())()

Metoda do zapisywanie się pliku.

(package private) static void

[**setUpServerConnection**](file:///C:\Users\kowal\Desktop\ProjektIPJAVA\doc\com\company\Main.html#setUpServerConnection())()

Metoda służy do połączenie z serwerem zewnętrznym, w którym są przechowywane wszystkie pliki.

(package private) static void

[**uploadFile**](file:///C:\Users\kowal\Desktop\ProjektIPJAVA\doc\com\company\Main.html#uploadFile(java.io.File,java.lang.String))([**File**](https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/io/File.html) pathSource, [**String**](https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/String.html) pathDestination)

Metoda służy do wysłania plików na serwer.

Metody klasy DocumentPreview:

(package private) static void

[**arrowKeysListener**](file:///C:\Users\kowal\Desktop\ProjektIPJAVA\doc\com\company\DocumentPreview.html#arrowKeysListener())()

metoda pozwalająca zmieniać strony wyświetlanego dokumentu za pomocą strzałek

(package private) static void

[**displayDocument**](file:///C:\Users\kowal\Desktop\ProjektIPJAVA\doc\com\company\DocumentPreview.html#displayDocument())()

metoda wyświetlająca dokument

 (package private) static void

[**poprzedniaStrona**](file:///C:\Users\kowal\Desktop\ProjektIPJAVA\doc\com\company\DocumentPreview.html#poprzedniaStrona())()

metoda zmieniająca stronę wyswietlonego dokumentu o jedną wstecz

(package private) static void

[**nastepnaStrona**](file:///C:\Users\kowal\Desktop\ProjektIPJAVA\doc\com\company\DocumentPreview.html#nastepnaStrona())()

metoda zmieniająca stronę wyswietlonego dokumentu o jedną dalej

Metody klasy PasswordHash:

static [**String**](https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/String.html)

[**createHash**](file:///C:\Users\kowal\Desktop\ProjektIPJAVA\doc\com\company\PasswordHash.html#createHash(char%5B%5D))(char[] password)

Returns a salted PBKDF2 hash of the password.

static [**String**](https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/String.html)

[**createHash**](file:///C:\Users\kowal\Desktop\ProjektIPJAVA\doc\com\company\PasswordHash.html#createHash(java.lang.String))([**String**](https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/String.html) password)

Returns a salted PBKDF2 hash of the password.

private static byte[]

[**fromHex**](file:///C:\Users\kowal\Desktop\ProjektIPJAVA\doc\com\company\PasswordHash.html#fromHex(java.lang.String))([**String**](https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/String.html) hex)

Converts a string of hexadecimal characters into a byte array.

private static byte[]

[**pbkdf2**](file:///C:\Users\kowal\Desktop\ProjektIPJAVA\doc\com\company\PasswordHash.html#pbkdf2(char%5B%5D,byte%5B%5D,int,int))(char[] password, byte[] salt, int iterations, int bytes)

Computes the PBKDF2 hash of a password.

private static boolean

[**slowEquals**](file:///C:\Users\kowal\Desktop\ProjektIPJAVA\doc\com\company\PasswordHash.html#slowEquals(byte%5B%5D,byte%5B%5D))(byte[] a, byte[] b)

Compares two byte arrays in length-constant time.

private static [**String**](https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/String.html)

[**toHex**](file:///C:\Users\kowal\Desktop\ProjektIPJAVA\doc\com\company\PasswordHash.html#toHex(byte%5B%5D))(byte[] array)

Converts a byte array into a hexadecimal string.

static boolean

[**validatePassword**](file:///C:\Users\kowal\Desktop\ProjektIPJAVA\doc\com\company\PasswordHash.html#validatePassword(char%5B%5D,java.lang.String))(char[] password, [**String**](https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/String.html) goodHash)

Validates a password using a hash.

static boolean

[**validatePassword**](file:///C:\Users\kowal\Desktop\ProjektIPJAVA\doc\com\company\PasswordHash.html#validatePassword(java.lang.String,java.lang.String))([**String**](https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/String.html) password, [**String**](https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/String.html) goodHash)

Validates a password using a hash.