* 1. Słownik
     1. Aplikacja - określa całość programu, jego interfejsu oraz kodu
     2. Użytkownik – osoba korzystająca z aplikacji
     3. Menu – określa obszar, w którym będą znajdowały się opcje do zarządzania aplikacją
     4. Opcja – pojedynczy przycisk znajdujący się w menu
     5. Katalog – główny katalog poddany współdzieleniu
     6. Drzewo katalogów – obszar, w którym będą wylistowane inne katalogi oraz pliki znajdujące się we współdzielonym katalogu
     7. Plik – pojedynczy element (niebędący katalogiem) znajdujący się w obszarze współdzielonym
     8. GUI – graficzny interfejs użytkownika
     9. Listener – urządzenie nasłuchujące i czekające na jakikolwiek pakiet od serwera
     10. Broadcaster – urządzenie wysyłające pakiet Discovery, żądające synchronizacji
     11. Pakiety:
         1. UDP port 1337
* **Sync request** – pakiet UDP zwierający:
  + jeden bajt danych z liczba 255 (1111 1111 binarnie) wysyłany do każdego hosta na liście w celu powiadomienia o chęci synchronizacji
  + numer sekwencyjny (32 bit liczba całkowita bez znaku) – będący kolejnym numerem pakietu który został wysłany
  + Tablicę znaków UTF8 z nazwą komputera (dowolna wielkość – do końca pakietu)
* **Sync notyfy –** pakiet UDP zwierający;
  + jeden bajt danych z liczba 170 (1010 1010 binarnie)
  + numer sekwencyjny (32 bit liczba całkowita bez znaku) – będący kolejnym numerem pakietu który został wysłany
  + Tablicę znaków UTF8 z nazwą komputera (dowolna wielkość – do końca pakietu)
  + wysyłany do każdego hosta na liście w celu powiadomienia o zmianie w lokalnych plikach
* **Duck ACK –** pakiet UDP zwierający:
  + jeden baj danych z liczba 0
  + numer sekwencyjny (32 bit liczba całkowita bez znaku) – taki sam jak dla pakietu na który jest odpowiedzią
  + Tablicę znaków UTF8 z nazwą komputera (dowolna wielkość – do końca pakietu)
  + pakiet jest potwierdzeniem otrzymania pakietów request lub notyfy
* **Discovery –** pakiet zawiera:
  + Liczbę 240 (1111 0000 binarnie)
  + adres ip w postaci liczby Int64 (big-endian) (pierwsze 8 bajtów)
  + Tablicę znaków UTF8 z nazwą komputera (dowolna wielkość – do końca pakietu) – powinna być unikalna - na końcu dodane jest losowe 8 znaków.
    - 1. TCP port 1338
* **Sync info packet** – zawiera:
  + Liczbę 255 (1111 1111 binarnie) (1 bajt)
  + liczbę pakietów elementów w głównym folderze (pliki – pakiety **File Info,** foldery – pakiety **Folder Info,** usunięte pliki – pakiety **File delete –** maksymalnie 2^32 elementów w folderze(4 bajty)
* **File request packet –** zawiera:
  + Liczbę 240 (1111 0000 binarnie) (1 bajt)
  + Rozmiar pakietu (32 bit liczba całkowita bez znaku)
  + Dane odebrane z pakiecie **File info packet:**
  + Sumę kontrolną pliku, który ma zostać wysłany - CRC (4 bajty)
  + Datę modyfikacji pliku, który ma zostać wysłany w postaci liczby milisekund, które upłynęły od 1970 – dla strefy czasowej UTC (8 bajtów)
  + **Rozmiar pliku (ilosc bajtów) – maksymalny rozmiar pliku 4 GB (4 bajty)**
  + **Nazwę pliku (do końca pakietu)**
* **No request packet –** zawiera:
  + Liczbę 160 (1010 0000 binarnie) (1 bajt)
* **Folder info packet –** zawiera:
  + Liczbę 10 (0000 1010 binarnie) (1 bajt)
  + Rozmiar pakietu (32 bit liczba całkowita bez znaku)
  + liczbę pakietów elementów w danym folderze (pliki – pakiety **File Info,** foldery – pakiety **Folder Info,** usunięte pliki – pakiety **File delete**  **–** maksymalnie 2^32 elementów w folderze (4 bajty)
  + **Nazwę folderu (do końca pakietu)**
* **File info packet** – zawiera: ////powiązanie z Sync request
  + Liczbę 170 (1010 1010 binarnie) (1 bajt)
  + Rozmiar pakietu (32 bit liczba całkowita bez znaku)
  + Sumę kontrolną pliku - CRC (4 bajty)
  + Datę modyfikacji pliku w postaci liczby milisekund, które upłynęły od 1970 – dla strefy czasowej UTC (8 bajtów)
  + Rozmiar pliku (ilosc bajtów) – maksymalny rozmiar pliku 4 GB (4 bajty)
  + Nazwę pliku (do końca pakietu)
* **File delete packet - zawiera:**
  + - **Liczbę 0** **(0000 0000 binarnie) (1 bajt)**
    - **Rozmiar pakietu (32 bit liczba całkowita bez znaku)**
    - **Sumę kontrolną pliku - CRC (4 bajty)**
    - **Datę modyfikacji pliku w postaci liczby milisekund, które upłynęły od 1970 – dla strefy czasowej UTC (8 bajtów)**
    - **Rozmiar pliku (ilosc bajtów) – maksymalny rozmiar pliku 4 GB (4 bajty)**
    - **Nazwę pliku (do końca pakietu)**
* **File packet –** zawiera:
  + dane binarne pliku (do końca pakietu)
  1. Typy danych
     1. String – typ tekstowy o dowolnej długości
     2. int – typ całkowity o długości 4 bajtów
     3. …

1. Specyfikacja wymagań
   1. Wymagania funkcjonalne
      1. Aplikacja jest napisana w języku obiektowym wysokiego poziomu Java SE 7? W środowisku ….
      2. Implementacja komunikacji
         1. Do synchronizacji katalogów pomiędzy różnymi platformami użyty jest algorytm Rsync lub piszemy własny od podstaw (do wyboru)
         2. Program ma sprawdzać aktualność danych automatycznie działając w tle w odstępach czasowych
         3. W przypadku niezgodności plików, aplikacja automatycznie dokonuje synchronizacji plików oraz katalogów
         4. Na podstawie synchronizacji tworzony jest harmonogram w celu późniejszej, ewentualnej weryfikacji
         5. Synchronizacja dokonywana także manualnie za pomocą przycisku „Synchronizuj”. Synchronizacja wykonywana jest przez użytkownika.
         6. Aplikacja identyfikuje użytkownika za pomocą wspólnego klucza.
         7. W przypadku niezgodności plików aplikacja wyświetla komunikat z pytaniem, który plik zapisać
         8. Usuwanie plików (2 tryby pracy aplikacji):
            1. Tryb z kasowaniem plików
            2. Tryb bez możliwości usuwania plików
         9. Plik nienadpisany lub usunięty zostaje umieszczony w katalogu „TEMP” w celu późniejszego ewentualnego odzyskania danych. Plik przechowywany 24 godziny od momentu usunięcia lub nienadpisania.
         10. W przypadku backup’u desygnowana jest do tego jedna maszyna. Tworzony jest harmonogram backup’ów.
         11. Harmonogramy synchronizacji oraz backup’u zapisywane w bazie danych
         12. W drzewie katalogów wyświetlane nazwy plików, waga oraz data ostatniej modyfikacji
         13. Przeniesienie pliku do współdzielonego katalogu jest w rzeczywistości jego kopią. Oryginalny plik zostaje również w poprzednim folderze ???
      3. Implementacja GUI
      4. Implementacja protokołu komunikacji
         1. Kodowanie znaków za pomocą systemu UTF-8
         2. Komputery są identyfikowane na podstawie nazwy – powinna więc być unikalna (można dla zwiększenia szybkości przeszukiwania listy hostów zastosować funkcją haszującą).
         3. Jednocześnie synchronizować mogą się tylko dwa urządzenia, natomiast oba urządzenia powinny jednocześnie wydawać oraz obsługiwać żądanie synchronizacji.
         4. Każde urządzenie wysyła co 1000 ms pakiet Discovery na adres Broadcast (Powinno zostać zrealizowane w osobnym wątku).
         5. Każde urządzenie w posiada cały czas jeden port UDP nasłuchujący na adres Anycast (powinno zostać zrealizowane w osobnym wątku), na porcie 1337.
         6. Po odebraniu pakietu Discovery, urządzenie sprawdza poprawność uzyskanych danych (adres ip i nazwę jak w pkt 2.1.4.1, w celu sprawdzenia czy nie jest to pakiet od samego siebie).
            1. Host którego dane zostały odebrane, dodawany jest do listy hostów.
            2. Jeśli istnieje host o takiej samej nazwie, **a innym adresie IP,** usuwany jest on z listy hostów.
         7. Strona kliencka wysyła żądanie synchronizacji:
            1. Urządzenie żądające synchronizacji z innym urządzeniem znajdującym się na liście urządzeń (strona kliencka) wysyła pakiet Sync request.
            2. Strona serwera oczekuje na pakiet Duck ACK
            3. Po odebraniu potwierdzenia Duck ACK tworzone jest nowe gniazdo TCP - adres IP urządzenia z listy, port 1338.
            4. Pomiędzy serwerem i klientem następuje wymiana kluczy TLS i autoryzacja.
         8. Synchronizacja:
            1. Strona serwera oczekuje na pakiet Sync info, następnie pakiety Folder info, File Info, Delete Info - do momentu otrzymania pakietu No request packet.
            2. **Dla każdego odebranego pakietu File delete packet:**

**Usunięcie plików dla których data modyfikacji pliku lokalnego jest wcześniejsza bądź równa dacie modyfikacji w otrzymanym pakiecie, lub gdy sumy kontrolne pliku są równe sumie kontrolnej w otrzymanym pakiecie.**

**Gdy data modyfikacji pliku lokalnego jest późniejsza od daty modyfikacji w otrzymanym pakiecie – zachowanie lokalnej kopii pliku**

* + - * 1. Porównanie otrzymanych informacji o plikach z informacjami o plikach lokalnych i podjęcie decyzji o synchronizacji plików:

**Otrzymano pakiet file info dla pliku który został usunięty na urządzeniu:**

**Gdy data modyfikacji pliku lokalnego, który został usunięty, jest późniejsza od daty modyfikacji w otrzymanym pakiecie :**

**w zależnośći od ustawień – odrzucenie pliku lub synchronizacja pliku i zachowanie go pod nazwą NAZWA\_ORG~NAZWA\_URZĄDZENIA**

Urządzenia A(działa w trybie serwera) i B(działa w trybie klienta) porównują sumy kontrolne.

Jeżeli sumy się zgadzają, nie dochodzi do synchronizacji.

W momencie, gdy sumy kontrolne są różne to:

Jeżeli plik w komputerze A ma starszą datę modyfikacji od pliku w komputerze B -> synchronizacja.

Jeżeli daty modyfikacji plików są wcześniejsze od daty ostatniej synchronizacji między komputerami -> przeniesienie do folderu temp i przetrzymanie do momentu decyzji użytkownika.

Jeżeli plik w komputerze B ma starszą datę modyfikacji od pliku w komputerze A –> brak synchronizacji.

* + - * 1. Dla każdego zsynchronizowanego pliku następuje wysłanie pakietu File request przez stronę kliencką.
        2. Klient oczekuje na pakiety z danymi od serwera.
        3. Klient sprawdza sumę kontrolną.
        4. Wysłanie pakietu No request
        5. Zamknięcie gniazda przez klienta.
      1. Strona serwerowa wysyła żądanie synchronizacji
         1. Po odebraniu pakietu Sync request przez Listnera serwer wysyła pakiet Duck ACK
         2. Serwer tworzy gniazdo TCP nasłuchujące na adres urządzenia, które wysłało pakiet Sync request. Port gniazda – 1339.
         3. Pomiędzy serwerem i klientem następuje wymiana kluczy TLS i autoryzacja.
         4. Synchronizacja:

Serwer wysyła pakiet **Sync info**, z informacją o synchronizowanych plikach na urządzeniu.

Dla każdego synchronizowanego folderu wysyłany jest pakiet **Forder info**.

Dla każdego synchronizowanego pliku w folderze wysyłany jest pakiet **File info**.

**Dla każdego usuniętego pliku w folderze wysyłany jest pakiet File Delete packet**

Serwer wysyła pakiet **No request packet**.

Serwer oczekuje na pakiet **File request** lub **No request**

Jeżeli **No request** -> zamknięcia gniazda

Jeżeli **File request** -> wysłanie **File packet** z danymi plików

* + - * 1. Po odebraniu pakietu Sync notyfy przez Listenera wysyłany jest w odpowiedzi pakiet Duck ACK i dochodzi do wysłanie plików z danymi.
      1. Obsługa błędów
         1. Jeżeli połączenie zostanie przerwane w trakcie synchronizacji podczas przesyłania danych binarnych pliku (File packet) – dane odbieranego pliku są porzucane.
         2. Jeżeli połączenie zostanie przerwane w trakcie synchronizacji podczas wymiany informacji o plikach (pakiety TCP z wyjątkiem File packet), następuje porzucenie odebranych informacji.
         3. Pliki które zostały zsynchronizowane w całości zostają uznane za zsynchronizowane.
         4. Jeżeli suma kontrolna CRC pliku nie zgadza się z sumą kontrolną otrzymaną w pakiecie File info dochodzi do ponownej wymiany plików.
         5. Jeżeli serwer lub klient wysłał jeden z pakietów Sync request, Sync notyfy, a nie otrzymano pakietu Duck ACK następuje ponowne wysłanie pakietu, dla którego nie otrzymano potwierdzenia po 1200 milisekundach.
      2. Każdy komputer posiada listę komputerów z którymi się synchronizował wraz z datą ostatniej synchronizacji

1. Decyzja o synchronizacji pliku na użądzeniu A(działa w trybie Serwera) i B(działa w trybie Klienta)
   1. Następuje porówanie sum kontrolnych plików
      1. jeżeli są takie same, brak sychronizacji
      2. jeżeli są różne:
         1. Jeżeli plik w komuterze A ma starszą date modyfikacji od pliku w komputerze B – Synchronizacja plików.
         2. Jeżeli daty modyfikacji plików są młodsze od daty ostatniej synchronizacji między komuterami **– oba pliki zostają zapisane na urzadzędzeniu pod nazwami NAZWA\_ORG~NAZWA\_URZĄDZENIA1 oraz NAZWA\_ORG~NAZWA\_URZĄDZENIA2.**
         3. Jeżeli plik w komuterze B ma starszą date modyfikacji od pliku w komputerze A – Brak synchronizacji plików.
2. **Usuwanie pliku**
   1. **Zapisanie informacji o usunietym pliku do lokalnego repozytoruim (np. baza danych) :**
      * **Sumę kontrolną pliku - CRC (4 bajty)**
      * **Datę modyfikacji pliku w postaci liczby milisekund, które upłynęły od 1970 – dla strefy czasowej UTC (8 bajtów)**
      * **Rozmiar pliku (ilosc bajtów) – maksymalny rozmiar pliku 4 GB (4 bajty)**
      * **Nazwę pliku**

* **Dane te powinny być przedtrzywane przez określoną przez użytkownika ilość czasu – domyślnie 14 dni**
  1. **Usunięcie pliku**

Kolejność wysyłania pakietów TCP – algorytm pseudokod:

* Wysłanie pakietu Sync info
* SendFolderInfo(rootFolder)

SendFolderInfo(Folder folder){

foreach ( File file in folder) {

sendFileInfoPacket(file);

}

foreach ( Folder subFolder in folder) {

SendFolderInfo(SendFolderInfo);

}

}

Kodowanie znaków – w tej wersji UTF8, ale może zmienić na coś mniejszego (ASCII??)

Odbieranie danych binarnych – pliki są podzielone na pakiety o rozmiarze od 1 bajtu do 10 Mib (Mebibajt – 2^20 bajtów) – o rozmiarze pakietu decyduje nadawca.  
Odbiorca oczekuje w sumie na N bajtów, gdzie N jest rozmiarem pliku z pakietu **File info packet,** poszczególne pakiety wysyłane są w kolejności w jakiej występują w pliku źródłowym.

Informacje o plikach (pakiety file info, folder info, sync info) powinny być aktualizowane po aktualizacji plików – w razie synchronizacji tylko pobierane a nie wyliczane.