Interface Homme-Machine
TP S04

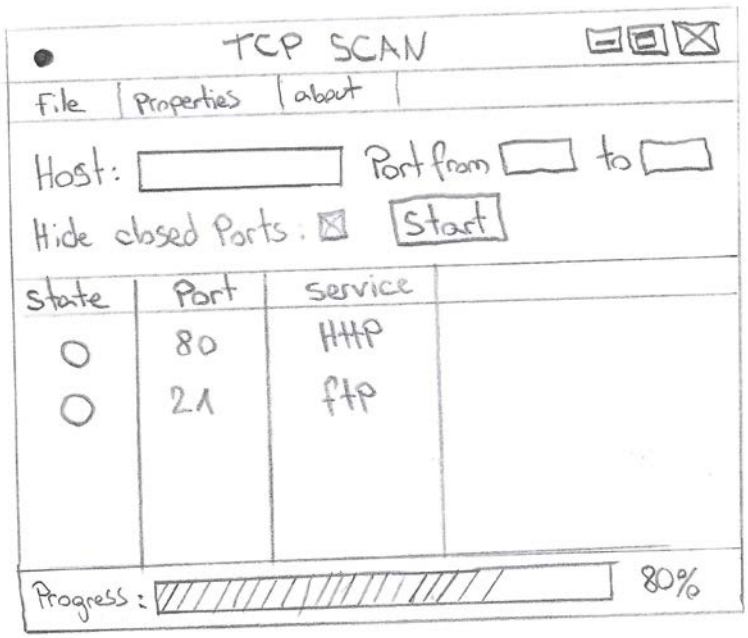
Date du travail :	16.06.2015	Etudiants :	Benoît Repond
Classe :	T-1a		Gilles Waeber

TCP Scanner

Table des matières

TCP Scanner

Maquette



Aperçu de l'application

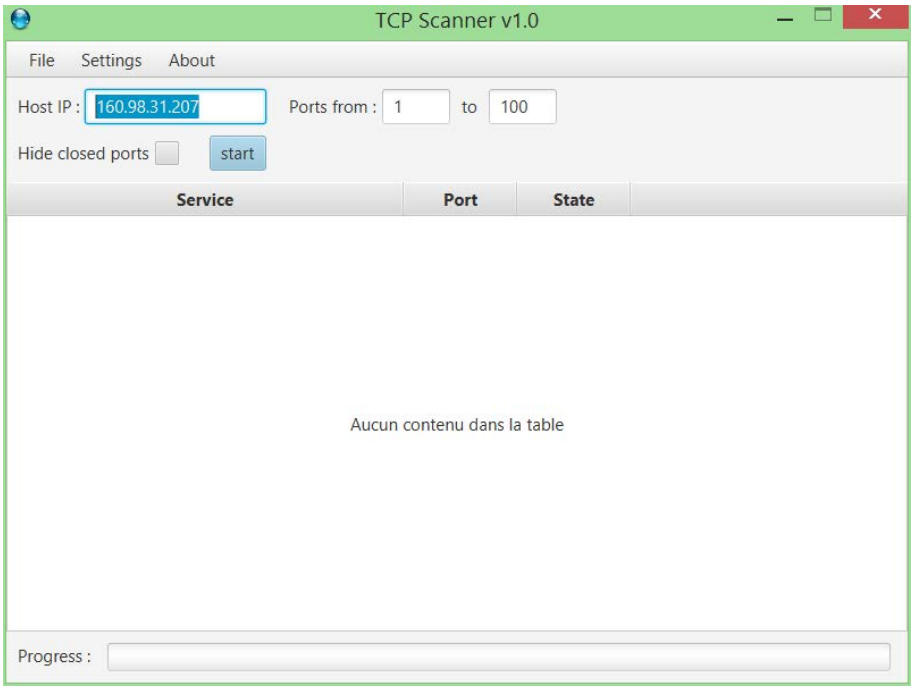


Figure 1 - Application au démarrage

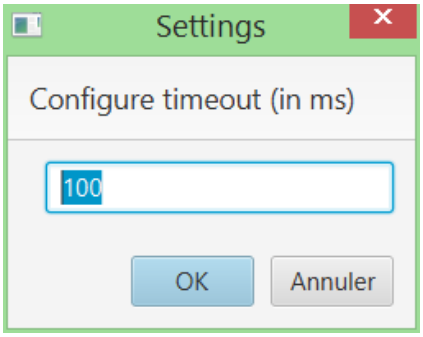


Figure 2 - Configuration du timeout (sous Settings)

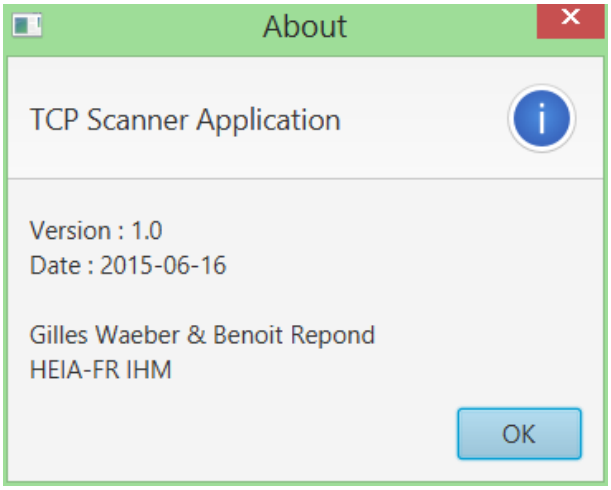


Figure 3 - Pop-up "A propos"

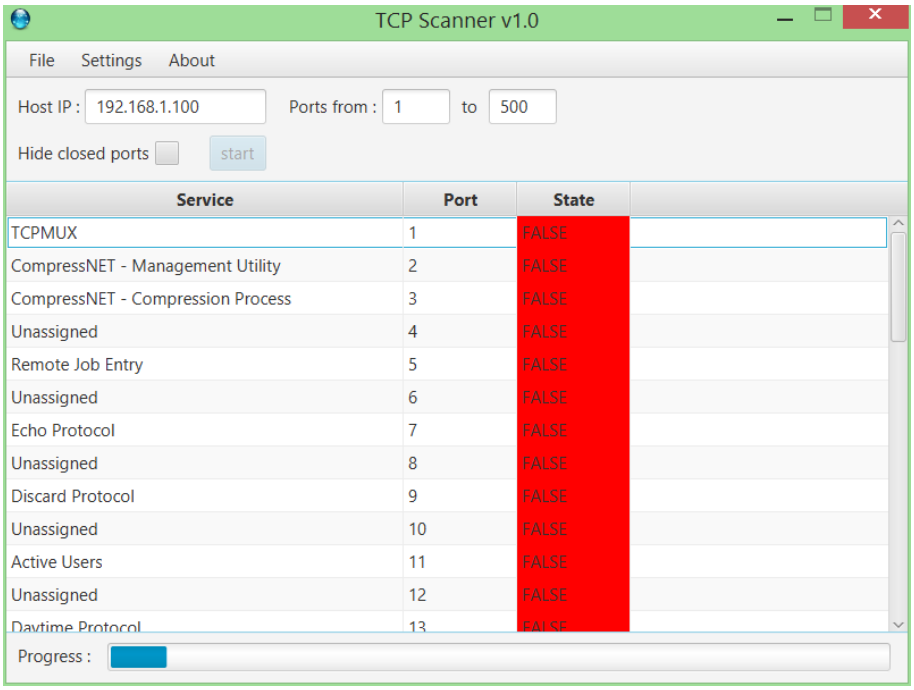


Figure 4 - Scan en cours

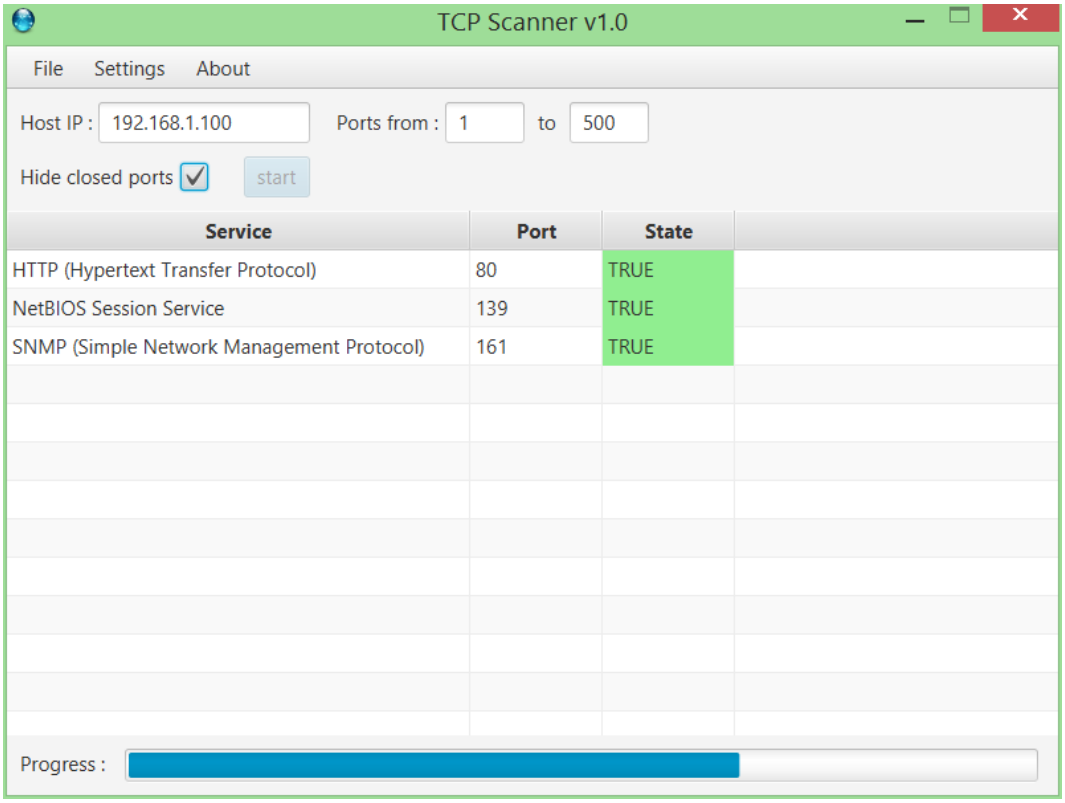


Figure 5 - Scan avec filtrage des ports non ouverts

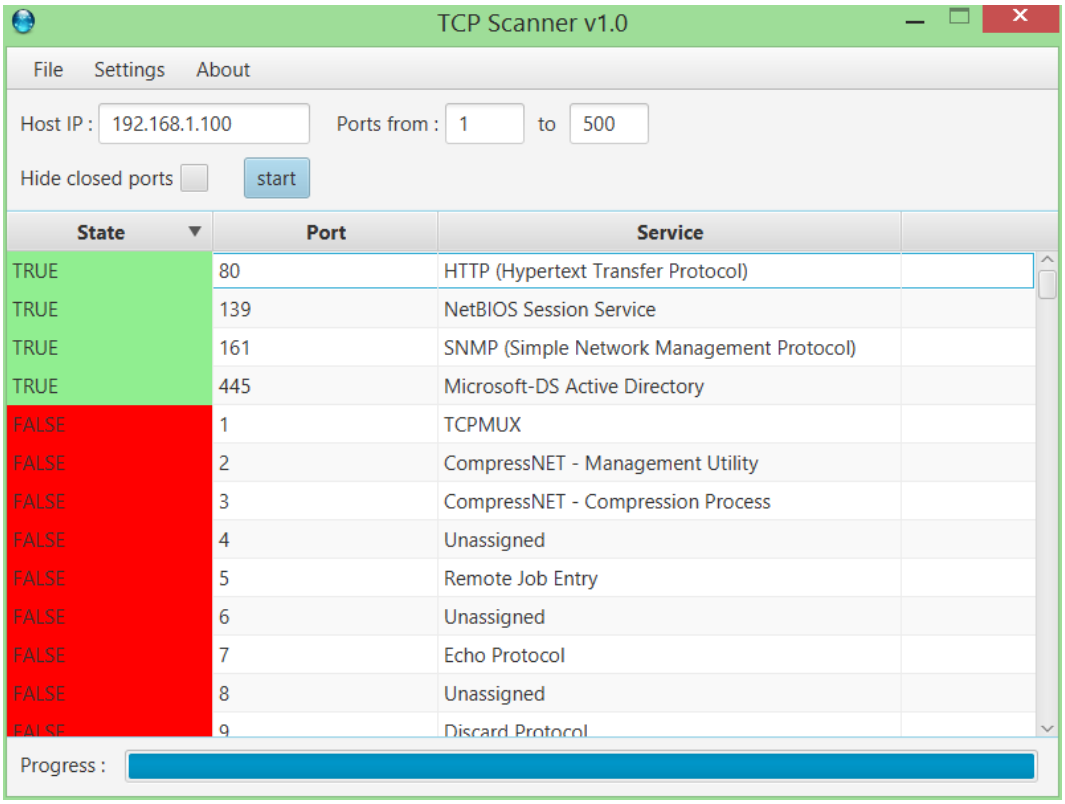


Figure 6 - Manipulation et filtrage des colonnes du tableau

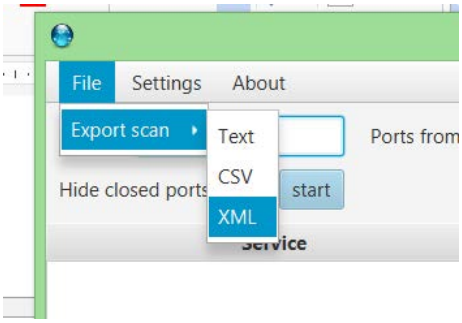


Figure 7 - Export des résultats

```
<?xml version="1.0" encoding="UTF-8"?>
- <tcpscan>
  - <port id="80">
      <state>true</state>
      <service>HTTP (Hypertext Transfer Protocol)</service>
    </port>
  - <port id="139">
      <state>true</state>
      <service>NetBIOS Session Service</service>
    </port>
  - <port id="161">
      <state>true</state>
      <service>SNMP (Simple Network Management Protocol)</service>
    </port>
  - <port id="445">
      <state>true</state>
      <service>Microsoft-DS Active Directory</service>
    </port>
</tcpscan>
```

Figure 8 - Export en XML

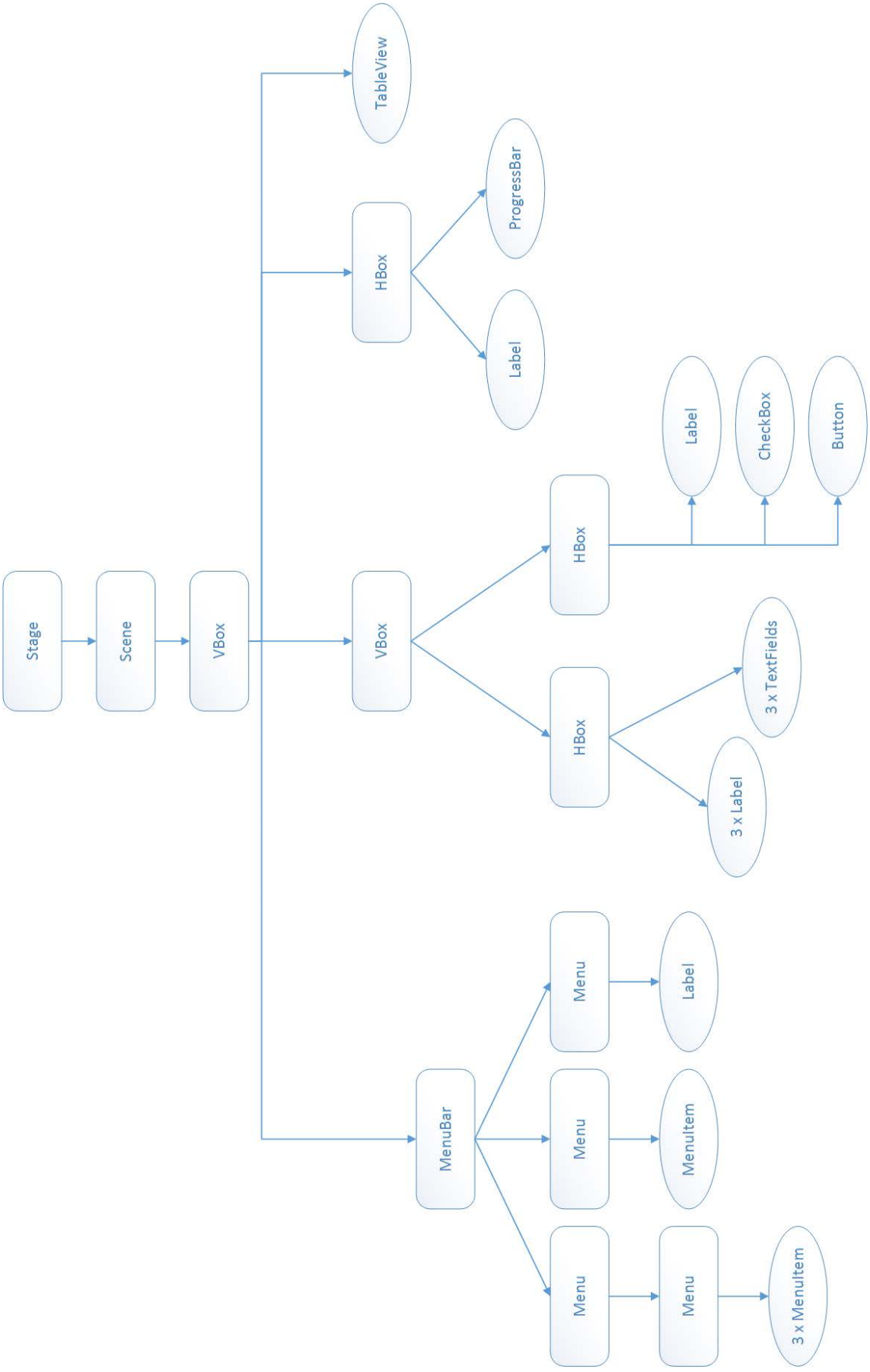
	A	B	C	D	E
1	port	state	service		
2	80	true	HTTP (Hypertext Transfer Protocol)		
3	139	true	NetBIOS Session Service		
4	161	true	SNMP (Simple Network Management Protocol)		
5	445	true	Microsoft-DS Active Directory		

Figure 9 - Export en CSV

Difficultés rencontrées

Nous avons essayés de faire du multithreading pour le scan de ports mais nous avons rencontré pas mal de problèmes (surtout au niveau de l'enregistrement des données du scan). Par manque de temps et de connaissances nous avons donc implémenté un simple thread.

Graphe de scène



Parties importantes du code

Thread pour les scans

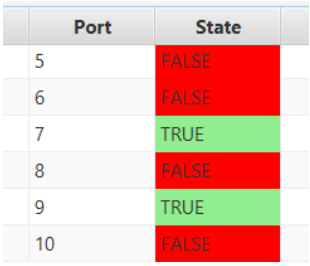
Pour réaliser le scan des ports TCP nous avons utilisé un thread pour faire tourner le scan en parallèle au reste de l'application. L'utilisateur peut donc continuer à utiliser l'application (filtrage des ports ouverts, manipulation du tableau, etc.) durant le scan.

```
package s04;

import java.net.InetSocketAddress;
import java.net.Socket;

public class TcpScannerScanThread extends Thread {

  private int firstPort;
  private int lastPort;
  private int timeOut;
  private String ip;
  private TcpScannerModel model;
  private TcpScannerView view;

  public TcpScannerScanThread(int firstPort, int lastPort, int timeOut, String ip,
TcpScannerModel model, TcpScannerView view) {
    this.firstPort = firstPort;
    this.lastPort = lastPort;
    this.timeOut = timeOut;
    this.ip = ip;
    this.model = model;
    this.view = view;
  }

  // Run a TCP scan
  public void run() {

    int first = firstPort;
    while(firstPort <= lastPort){

      String service = model.getServiceByPort(firstPort);
      boolean state = portIsOpen(ip, firstPort, timeOut);
      model.addPort(new Port(state, firstPort, service));

      double progress = (double)(firstPort-first) / (double)(lastPort-first);
      view.updateProgress(progress);

      firstPort++;

    }

    view.disableStartButton(false);

  }

  // Test if port is open
  public static boolean portIsOpen(String ip, int port, int timeout) {

    try {

      Socket socket = new Socket();
      socket.connect(new InetSocketAddress(ip, port), timeout);
      socket.close();
      return true;

    } catch (Exception e) {

      return false;

    }

  }

}
```

Tableau

Pour afficher les résultats d'un scan nous avons utilisé le composant TableView qui permet de mettre en forme des données sous la forme d'un tableau dynamique (filtrage et déplacement des colonnes par exemple).

Nous avons personnalisé notre tableau pour que les cellules de la colonne statu prennent automatiquement un fond rouge ou vert suivant le statu du port.



Port	State
5	FALSE
6	FALSE
7	TRUE
8	FALSE
9	TRUE
10	FALSE

```
    TableColumn serviceCol = new TableColumn("Service");
    serviceCol.setMinWidth(350);
    serviceCol.setCellValueFactory(new PropertyValueFactory<Port, String>("service"));

    TableColumn portCol = new TableColumn("Port");
    portCol.setMinWidth(100);
    portCol.setCellValueFactory(new PropertyValueFactory<Port, Integer>("port"));

    TableColumn stateCol = new TableColumn("State");
    stateCol.setMinWidth(100);
    stateCol.setCellValueFactory(new PropertyValueFactory<Port, Boolean>("state"));

    stateCol.setCellFactory(new Callback<TableColumn, TableCell>() {
      public TableCell call(TableColumn param) {
        return new TableCell<Port, Boolean>() {

          public void updateItem(Boolean item, boolean empty) {
            super.updateItem(item, empty);
            if (!isEmpty()) {

              if(item){
                this.setBackground(new Background(new BackgroundFill(Color.LIGHTGREEN,
CornerRadii.EMPTY, Insets.EMPTY)));
                setText("TRUE");
              }else{
                this.setBackground(new Background(new BackgroundFill(Color.RED,
CornerRadii.EMPTY, Insets.EMPTY)));
                setText("FALSE");
              }

            }else{
              this.setBackground(new Background(new BackgroundFill(null, null, null)));
              setText(null);
            }

          }
        };
      }
    });

    table.setEditable(true);
    table.setItems(model.getPortOList());
    table.getColumns().addAll(serviceCol, portCol, stateCol);
    table.setPadding(new Insets(0, 0, 0, 0));
```

Test utilisateur

Scénario de test

Points à évaluer

- Tâches réussies : Comptabiliser le nombre de tâches réussies par rapport à celles qui sont demandées dans le scénario.
- Temps d'exécution : Mesurer le temps nécessaire pour se familiariser avec l'application et effectuer les tâches (temps total - temps de scan).
- Erreurs commises : Mesurer les erreurs commises par l'utilisateur.

Script du test

L'utilisateur doit trouver les ports TCP ouverts d'une machine donnée (adresse IP) en utilisant l'application TCP Scanner. Le scan doit être effectué avec un timeout de 120ms et se limitera aux 1024 premiers ports.

Une fois qu'il a trouvé les ports TCP ouverts, il doit exporter les résultats dans 2 fichiers :

- La liste de tous les ports scannés en .xml
- La liste des ports ouverts en .csv

Tâches à effectuer / à contrôler :

1. « Configurer » l'adresse IP de la machine cible
2. « Configurer » correctement un range de ports (0 à 1024)
3. Configuration du timeout
4. Export en .xml
5. Filtrage des ports ouverts seulement
6. Export des ports ouverts en .csv

Question à poser à la fin du test :

- L'application est facile à prendre en main sans support d'aide.
- Il faut peu d'actions pour réaliser un scan de ports.
- Les messages d'informations / d'erreurs sont clairs et compréhensibles.
- La mise en forme du contenu dans un tableau est agréable et facilite la lecture des résultats.

Echelle de réponses : Pas du tout d'accord
Plutôt pas d'accord
Moyennement d'accord
Plutôt d'accord
Totalement d'accord

Analyse des résultats

Remarque : Etant donné que l'application n'est pas destinée au grand public, le test a été effectué sur 5 personnes travaillant dans le domaine IT.

Résultats du test

- Tâches réussies : Chaque participant a réussi entièrement les 6 tâches à effectuer.

- Temps d'exécution :
  Les utilisateurs ont tous effectués le test entre 2 minutes 40 et 3 minutes. Un scan de 1024 ports dure un peu plus de 2 minutes. Nous en avons donc déduit que les utilisateurs ont pris entre 40 secondes et une minute pour se familiariser avec l'application, configurer les réglages et exporter les résultats.
  Pour une première « découverte » de l'application nous trouvons que ce temps est relativement bon et que l'application est suffisamment claire pour une prise en main rapide.

- Erreurs commises :
  - Un des participants à inversé l'ordre des ports mais l'application lui a immédiatement signalée le problème et il a pu de lui-même corriger son erreur.
  - Un des participants a fait une faute de saisie lors du remplissage de l'adresse IP et il a dû quitter l'application pour interrompre le scan.

Questions

Nous avons effectué une pondération (de 1 à 5) pour les réponses aux questions de notre scénario de test.

Questions	Moyennes
L'application est facile à prendre en main sans support d'aide.	4,8
Il faut peu d'actions pour réaliser un scan de ports.	4,8
Les messages d'informations / d'erreurs sont clairs et compréhensibles.	3,5
La mise en forme du contenu dans un tableau est agréable et facilite la lecture des résultats.	5

Synthèse des problèmes d'utilisabilité

Le test que nous avons effectué a permis de ressortir les deux problèmes suivants :

- Lorsqu'un scan est lancé, l'utilisateur n'a aucun moyen de l'arrêter, il est obligé de quitter l'application.
  Le critère ergonomique concerné est le *contrôle utilisateur* (dans contrôle explicite).
- Lorsqu'un paramètre est mal configuré, un message d'erreur est affiché. Ce message d'erreur concerne l'ensemble des paramètres (adresse IP et les 2 ports) même si seulement un seul paramètre est mal configuré.
  Le critère ergonomique concerné est la *qualité des messages d'erreurs*.

Proposition de modification / d'amélioration

En plus des fonctionnalités optionnelles qui n'ont pas été implémentées, les 2 points suivants doivent être améliorés :

- Affiner les informations affichées dans le message d'erreur lorsque des paramètres sont incorrects.
- Ajouter la possibilité d'arrêter un scan en cours. Pour ne pas surcharger l'interface, nous avons pensé implémenter cette fonction directement dans le bouton « Start » en le transformant en un bouton « Cancel » lorsqu'un scan est en cours.

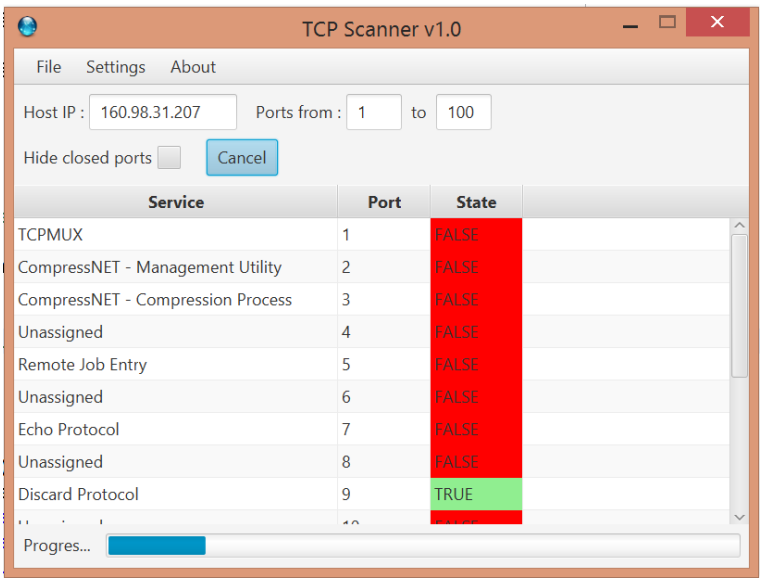


Figure 10 - Bouton cancel en cours de scan (proposition de modification)

- Trouver l'adresse IP d'une machine automatiquement si l'utilisateur saisie le nom d'hôte et pas l'adresse IP de la machine à scanner.

En plus des améliorations au niveau de l'interface et des fonctionnalités, les performances du scan peuvent être également améliorées. Actuellement les ports sont scannés un par un par un thread en dehors de l'application principale. Pour que le scan s'effectue plus rapidement, on pourrait modifier l'application pour effectuer le scan en multithreading.