**תיק פרוייקט**

רקע

תיאור המוצר המוגמר

שם המוצר הוא Network Manager Tool.

ביכולתו להדליק ולכבות מחשבים מרחוק (בתוך הרשת) לפי זמן שנקבע מראש. על מחשבים בהם פועלת תוכנת הלקוח אפשר לבצע פעולות נוספות: גישה לקבצים, מחיקתם או יצירת חדשים, גישה לתהליכים עצירתם או הפעלת תהליכים אחרים. ממשק המשתמש של התוכנה נפתח בדפדפן ויכול להפתח מכל מחשב ברשת בעזרת הדפדפן כל עוד השרת עובד באחד המחשבים ברשת.

המטרות המרכזיות של ה-Network Manager Tool הן הקלת העבודה על מנהל הרשת. הוא יכול לפקח על המחשבים ופעולותיהם מרחוק ואין לא צורך לגשת אליהם פיזית. בנוסף, הממשק מאפשר גישה לשרת מכל מחשב ברשת דרך הדפדפן כך שגם אם מנהל הרשת לא ליד העמדה שמחוברת למחשב השרת, כל עוד עמדתו מחוברת לרשת הוא יכול לגשת אליה. כל זה קורה תוך תקשורת מאובטחת בין השרת והלקוחות.

שאלות שהמערכת עונה עליהן:  
**כיצד ימצאו תוכנות הלקוח את השרת?** כל כמה שניות, השרת שולח הודעת Broadcast שכוללת את המפתח הפומבי שלו ו-Hash של הודעה שנקבע מראש. תוכנת הלקוח שהופעלה זה עתה תאזין לקשורת ותאתר את ההודע הזו, לאחר אימות של המסר שנקבע מראש, יתחיל הלקוח בתהליך החיבור לשרת.  
**כיצד להפעיל מרחוק מחשב?** קיימת אפשרות להפעיל מחשב מרחוק דרך כרטיס הרשת. יש לסדר בהגדרות המחשב שגם כאשר הוא כבוי, כרטיס הרשת ישתמש במעט חשמל כדי להאזין לתקשורת . כאשר כרטיס הרשת מוצא Magic Packet שמיועדת לו, הוא יפעיל את המחשב.

השוואת העבודה עם פתרונות ויישומים קיימים: ישנן תוכנות המבצעות Wake on LAN וגם תוכנות דרכן אפשר לפקח מרחוק על המחשבים ברשת. לא נמצאה תוכנה שלממשקה אפשר לגשת דרך הדפדפן, כאמור דבר המאפשר נוחות וגישה לשרת גם מהמחשבים שלא מריצים אותו עליהם. לעומת זאת בתוכנה הנוכחית, אין פיקוח על דברים הקשורים בחומרה כמו טמפרטורת מעבד.

מבט אישי על העבודה ותהליך הפיתוח

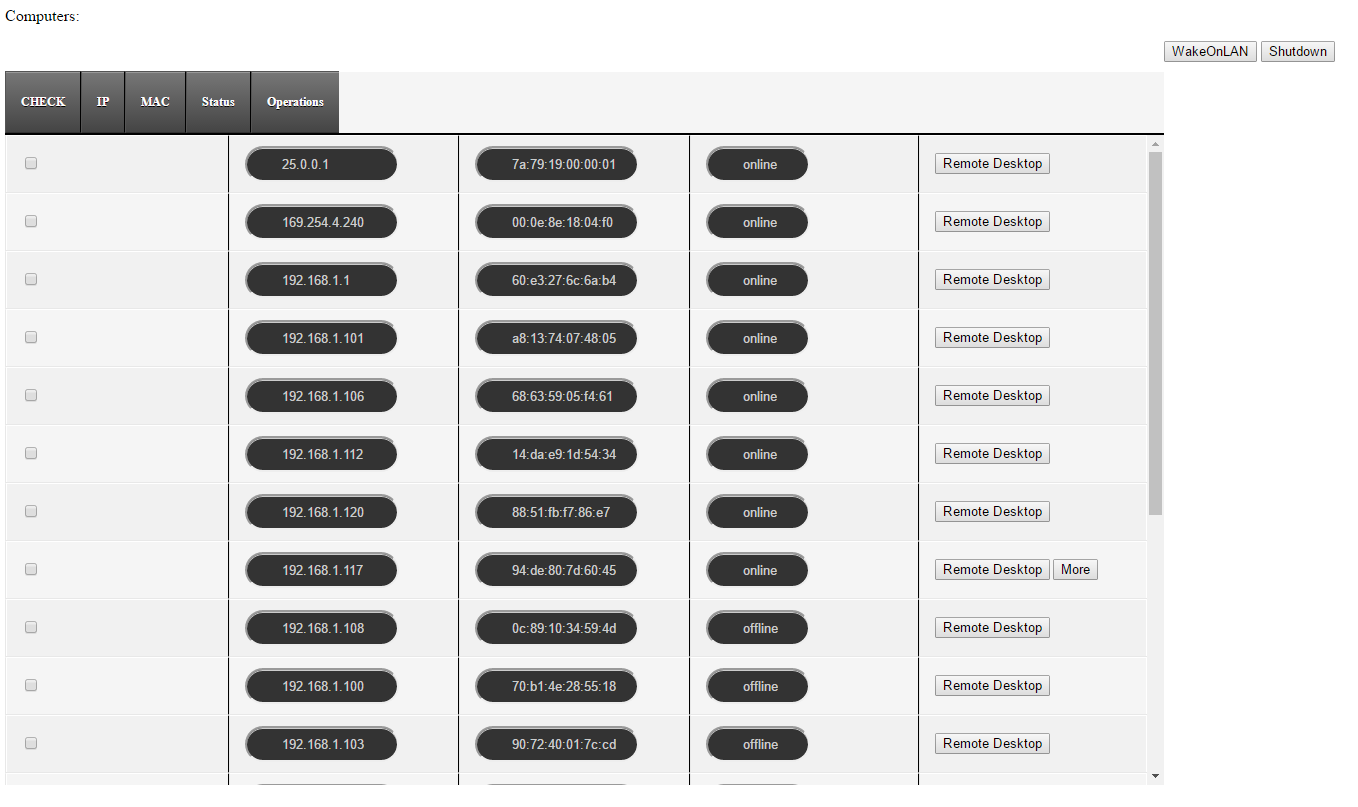
תיאור הממשק למשתמש

תחילה, יש להתקין את תוכנת השרת על מחשב אחד ברשת (לפחות) ואת תוכנת הלקוח על כל המחשבים.

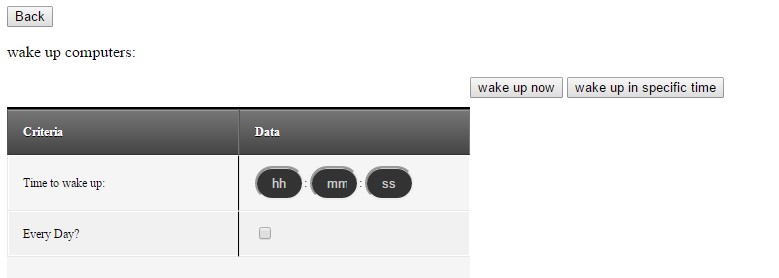
כדי לאפשר הדלקת מחשבים מרחוק, בכל מחשב יש לאפשר ב-BIOS את האפשרות ל-Wake on LAN או Wake on magic packet תחת הקטגוריה Power management.

לאחר שכל אלו נעשו, יש להפעיל את השרת. מייד יפתח חלון חדש או לשונית חדשה בדפדפן שתציג את ההודעה: Server is starting. It might take a while…. יש לחקות עד לעליית השרת.

החלון הראשי:

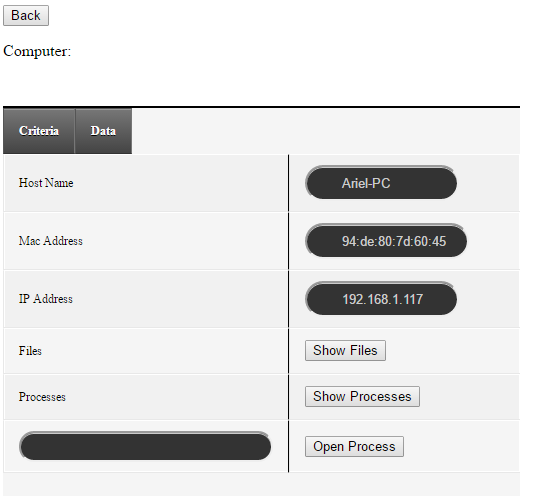


מציג טבלה של כל המחשבים ברשת שהשרת יודע על קיומם. אפשר לבחור מהם מחשבים ולהפעיל אותם על ידי לחיצה על כפתוח Wake on LAN שיביא למסך הזה:



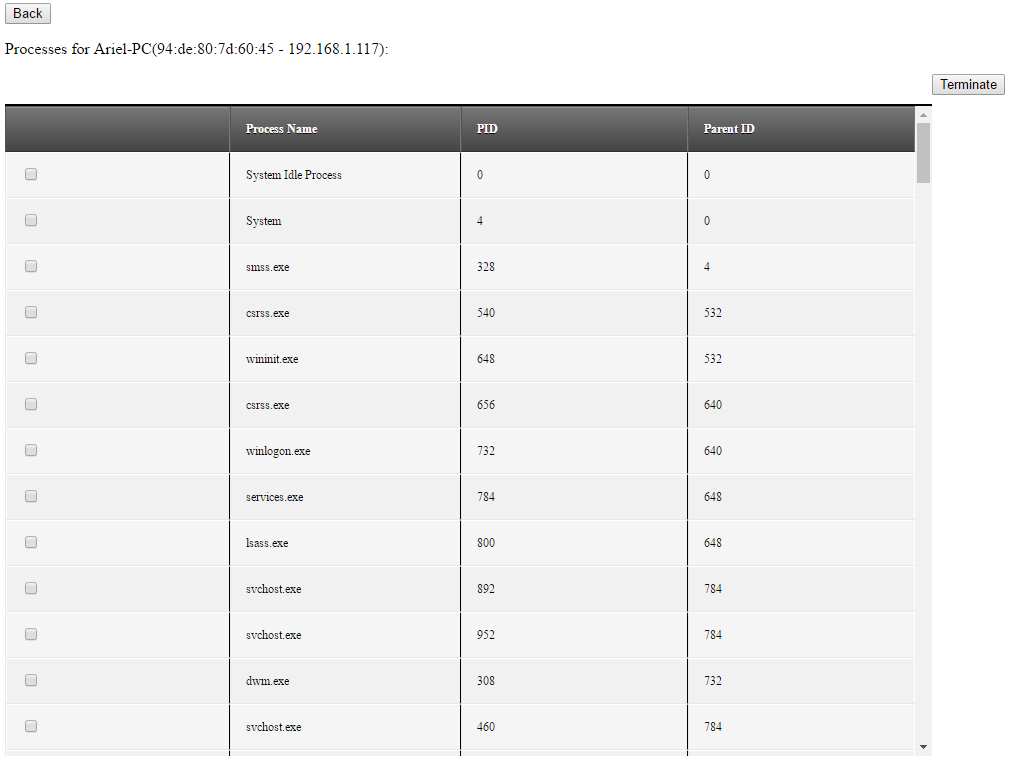
כאן אפשר לבחור אם להפעיל את המחשב בזמן ספציפי או מייד ולבחור את הזמן. אפשר גם לעשות שפעולה זו תחזור על עצמה על יום. לחיצה על Shutdown במסך הראשי תביא לחלון זהה.

ליד מחשבים שעליהם מופעלת תוכנת הלקוח יופיע כפתוח More, לחיצה עליו תביא למסך:



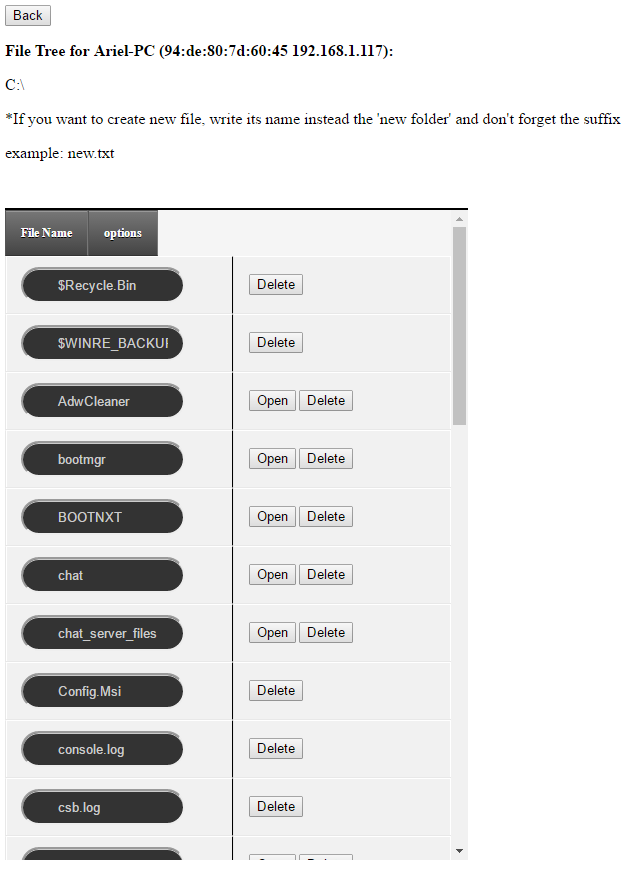
מסך זה מציג מידע על המחשב הספציפי הזה. כאן אפשר לכתוב פקודה שתרוץ בshell של המחשב ולשלוח אותה עם לחיצה על Open Process, אפשר לגשת לחלון התהליכים וחלון הקבצים על ידי לחיצה על Show Processes ו-Show Files בהתאם.

חלון תהליכים:



מציג את כל התהליכים הרצים על המחשב. אפשר לבחור מתוכם ואז ללחוץ על Terminate לסגירתם המיידית.

חלון הקבצים:

מציג את התוכלה של תיקיות. לחיצה על הכפתוח Open תציג את תכולת התיקייה. לחיצה על Delete תמחק את הקובץ או התיקייה לאלתר.

כדי לגשת לממשק המשתמש דרך הדפדפן יש לכתוב את כתובת ה-IP של המחשב עליו רץ השרת, נקודותיים ואז-5000. לדוגמה:



סביבת העבודה

הפרויקט נכתב ב-Python, ב- HTML וב-CSS. ליבת הפרויקט, ועיקרו נכתבו בשפה Python בשל הנוחות של השפה. ה-GUI נכתב ב-Python, ב-HTML וב-CSS. HTML ו-CSS שימשו לעיצוב ה-GUI עצמו כדי שיהיה אפשר להציג אותו בקלות ונוחות בדפדפן. הקוד שמחבר בין ה-GUI לתוכנה נכתב ב-Python. כדי לחבר בין הקוד שנכתב ב-Python לדפדפן שמשתמש בפרוטוקול HTTP, השתמשתי בספריה שהוכנה עבור Python בשם Flask. זו יוצרת שרת HTTP, מכסה את התקשורת איתו ומעניקה למתכנת כלים כדי להציג בקלות דפי HTML שהוא עיצב.

סביבת העבודה בה השתמשתי עבור שלוש השפות היא PyCharm. היא תוכנה מקצועית ונוחה לעבודה על פרויקטים מרובי קבצים ומאפשרת גם הרצה ו-debugging.

תיאור המודולים

המודולים המרכיבים את הפרוייקט והקשרים ביניהם:

* מודול GuiEngine. מודול זה מפעיל את ממשק המשתמש ואת מנוע השרת ומהווה את החוליה המקשרת ביניהם.
* מודול Server מהווה את תוכנת השרת העיקרית. הוא מופעל על ידי מודול GuiEngine ומתופעל חלקית על ידיו. מודול זה משתמש במודולים הבאים:
  + מודול ComputerDatabase שמהווה את הקישור של השרת למאגר הנתונים שלו.
  + מודול NetMap שתפקידו מיפוי הרשת ומציאת כל היישויות המחוברות בה.
  + מודול ClientInterface שמכסה את הממשק של השרת לקשר עם הלקוחות.
  + מודל Cipher שאחראי על ההצפנה ואבטחת המידע.
* מודול ClientInterface משתמש במודולים: WakeOnLan שמכסה את הפעלת המחשבים וכיבויים מרחוק, ובמודול Process שמכיל אובייקט לייצוג תהליך.
* מודול ComputerDatabase משתמש באובייקט Computer לייצוג מחשב ממודול ClientInterface.
* מודול Client מהוות את תוכנת הלקוח. הוא משתמש במודולים:
  + Process שמכיל את האובייקט Process.
  + Cipher
* כל המודולים משתמשים במודול Constants שמכיל את הקבועים של התוכנה.

תיעוד הקוד

הפרויקט משתמש במסד נתונים שבו טבלה אחת. טבלה זו, בשם Computers, מכילה מידע על כל המחשבים ברשת: כתובת Mac, כתובת IP (נכונה עבור העדכון האחרון) ומצב: online או offline (נכון לעידכון האחרון). כאשר השרת רץ, טבלה זו מתעדכנת כל דקה.

האלגוריתמים המרכזיים בפרוייקט

תדפיס של הקוד