פרויקט גמר לשנה"ל תשע"ה

Network Environment Awareness

הוכן לשם השלמת הדרישות לקבלת

תואר ראשון בהנדסה B.Sc

מאת

ולדימיר סלודייב

פבל קרס

מרגריטה גרינברג

דמיטרי פוזין

בהנחיית

ד"ר אורלוב מיכאל

בשיתוף ממר"ם (צה"ל)

הוגש למחלקה להנדסת תוכנה

המכללה האקדמית להנדסה סמי שמעון

באר שבע

תשע"ה

2015

אישור המנחה:_____

אישור ראש המחלקה:_____

תוכן עניינים

1. מבוא 4
1.1. מטרת הפרויקט 4
1.2. אופן ביצוע הפרויקט 5
2. מסמך דרישות SRS 6
2.1. יעדים 6
2.1.1. מטרת המסמך 6
2.1.2. משתמשי המערכת 6
2.1.3. קהל היעד 6
2.1.4. הנחות ואילוצי מערכת 6
2.2. תיאור כללי 6
2.2.1. תכלית הפרויקט 6
2.2.2. מאפייני התוכנה 6
2.2.3. סביבת עבודה 7
2.3. יעדים ומטרות 7
2.3.1. יעדים 7
2.3.2. מטרות 8
2.4. בעיות 8
2.5. יישום 8
2.5.1. סקירת מצב קיים 8
2.5.2. אופי המערכת וסוגה 8
2.5.3. אילוצי המערכת 8
2.5.4. מילון מונחים 9
2.5.5. ממשק תפעולי 10
2.5.6. .Class Diagram 10
2.5.7. סקירת ה– Use Case 11
2.5.8. מבנה המערכת 13
2.5.9. מאפייני המערכת 13
2.6. טכנולוגיה ותשתית 14
3. ניהול עבודה בצוות 15
3.1. מתודולוגיית פיתוח תוכנה 15
3.1.1. פיתוח תוכנה זריז (Agile) 15
3.1.2. SCRUM 15
3.2. תכנית עבודה בפרויקט 16
4. שפת הפיתוח : Java 19

4.1. Java Language 19
4.1.1. Java Virtual Machine 19
4.2. למה בחרנו בשפת תכנות Java 20
5. UI – (User Interface) 21
5.1. Swing 21
5.1.1. חלון הגדרת Check 21
5.1.2. חלון הגדרת Action 22
5.1.3. חלון הגדרת Policy 22
6. תיאור השיטה 23
6.1. Task Scheduler 23
6.2. אלגוריתמים לבדיקת רשת 23
6.3. Policy 24
7. UML 25
7.1. Sequence Diagrams 25
7.1.1. ביצוע בדיקות סביבת רשת 25
7.1.2. בחירת פעולות לפי Policy 25
7.1.3. ביצוע פעולות 26
8. ארכיטקטורת תוכנה 27
8.1. תבניות עיצוב (Design Patterns) 27
8.1.1. תבנית Factory Method 27
8.1.2. תבנית Thread Pool 28
9. בדיקות מערכת 30
9.1. מטרת הפרק 30
9.2. תיאור המערכת 30
9.3. שלב תכנון הבדיקות STP 30
9.3.1. אסטרטגיית הבדיקות 30
9.3.2. תיחום הבדיקות 30
9.3.3. דרישות לביצוע הבדיקות 30
9.4. STD+STP 31
9.4.1. בדיקות ממשקי משתמש בדיקות – GUI 31
10. המלצות לפיתוח עתידי 39
11. סקר ספרות 40
11.1. הקדמה 40
11.2. תיאור הבעיה 41
11.3. השיטה 41

11.4. בדיקות היכולות להעיד על חיבור לרשת זרה 41
11.5. זיהוי התחברות לרשת 42
11.6. הגנת המחשב במידה והמחשב ברשת זרה 43
12. ביבליוגרפיה 44

1. מבוא

אנו סבורים כי מידע הוא אלמנט יקר בארגון, דליפת מידע זה יכול להיות מסוכן לארגון.
כיום יש הרבה איומים לגניבת מידע מהארגון ולכן השאיפה היא למנוע מגורמים לא מורשים להשיג מידע זה.
בימים אלו, רוב הארגונים מנסים לפתור את הבעיה הזו בעזרת טכניקות שונות. יש הרבה אפשרויות לשמירת מידע בתוך הארגון כגון שרתים מאובטחים עם חיבור VPN , SMART CARDS , מערכות לניהול הרשאות וכדומה.
אנו סבורים כי חלק מהמידע החשוב שמור גם על מחשבים רגילים של המשתמשים בארגון, לכן רוב הניסיונות של הארגון להגנת מידע זה מדליפה לא יעזרו, כיוון שאין צורך להתחבר לשרת על מנת להשיג מידע זה. כמו כן לאחר חיבור למחשב בצורה תקינה, למשתמש יש גישה למידע החשוב לארגון ובזמן עבודתו מחשב יכול בטעות להתחבר לרשת אחרת ובשלב זה, יש סכנה לדליפת מידע.
בפרויקט שלנו אנו ננסה למצוא פתרון לבעיה זו ע"י דיווח ומניעה מהמשתמש להתחבר לרשת זרה.
הפרויקט יהיה מחולק לשתי חלקים עיקריים שהם :

- זיהוי התחברות לרשת זרה.
- פעולות שיש לנקוט בעת זיהוי חיבור לרשת זרה.

בפרויקט זה אנו מהווים חלק מצוות פיתוח של תוכנת עזר המותקנת על מערכת הפעלה ווינדוס שתפקידה ליישם את החלקים העיקריים של הפרויקט. אנו עושים פרויקט זה בשיתוף פעולה עם יחידה צבאית ממר"ם. תפקידנו ליצור אב טיפוס מודרני שיספק את מטרת הפרויקט.
על מנת ליישם את הרעיון אנו משתמשים בשירות של מערכת ההפעלה Task Scheduler.
שירות זה קיים בכל הגרסאות של ווינדוס החל Windows 98 לכן החלטנו להשתמש בתוסף זה כדי לזהות חיבורים או התנתקויות מרשת. בעת חיבור לרשת כלשהי השירות יפעיל את תוכנת העזר אשר תבצע סדרת בדיקות ופעולות על מנת לספק הגנה על מידע רגיש השמור על המחשב.

1.1. מטרת הפרויקט

מטרת העל של הפרויקט היא להגן על החומר הסודי של המחשב הצבאי, אך לשם כך יש צורך לזהות לפי קריטריונים שונים שאנו בהכרח ברשת זרה . לכן הפרויקט ישקיע מאמץ רב על מנת לזהות מצב בו המחשב מתחבר לרשת שלא אמור להיות בה. התוכנה תפיק דוח לגביי פרטים של הרשת הזרה, ותישלח פרטים אלו לגורמים הרלוונטיים בצבא, או תשמור תעבורת מידע של המחשב עצמו בצורה שקטה כדי לחקור בעתיד האם החיבור לרשת זרה היה בכוונה או בטעות .
בנוסף התוכנה תבצע פעולות שהוגדרו לה מראש על מנת לשמר את המידע החשוב לדוגמא: לכבות מחשב, לצאת מהמשתמש וכדומה.
הפרויקט שלנו יהווה שלד לתוכנה עתידית של הצבא, לכן המטרה העיקרית הנוספת היא לארגן את האב טיפוס כך שיהיה גמיש ונוח להוספת שיטות חדשות של בדיקות או פעולות.
אנו נראה ונסביר בפרויקט באיזה אופן בחרנו לממש כל אחד מהדברים ונסביר למה שיטה זו תיתן גמישות לפרויקט.

1.2. אופן ביצוע הפרויקט

1. מחקר אודות הנושאים הנכללים בפרויקט.
2. הכנת לוח זמנים מפורט של תהליכים בפרויקט.
3. איסוף שיטות למימוש.
4. כתיבת התוכנית ע"פ מודל Agile .
5. איתור חריגות.
6. בדיקות תוכנה בהתאם למסמכי דרישות.

2. מסמך דרישות SRS

2.1. יעדים

2.1.1. מטרת המסמך

מטרת המסמך היא לרכז את כל הנתונים הדרושים לקבלת כל ההחלטות הדרושות בפרויקט. מסמך זה מיועד למפתחים וללקוחות כאחד. כמו-כן, מסמך זה מספק מידע על ההתנהגות של המערכת, דרישותיה, אילוציה והגדרת הממשק מנקודת מבט של המשתמש וקהל היעד.

2.1.2. משתמשי המערכת

התוכנה נבנתה עבור אנשי בטיחות מידע ומנהלי רשת של הצבא אשר אמורים להתקין את התוכנה ולתת הגדרות התחלתיות על מנת שהתוכנה תוכל להתחיל לעבוד. התוכנה לא נועדה עבור משתמשים רגילים של המחשב אשר התוכנה מותקנת עליו, התוכנה תתחיל לרוץ בעת התחברות לרשת בצורה אוטומטית.

2.1.3. קהל היעד

מסמך זה מיועד ל :

- צוות פיתוח התוכנה .
- סגל מחלקת הנדסת תוכנה .
- כל אדם אשר יש לו עניין בפיתוח ושדרוג של מערכת .
- מסמך זה יציג את נקודות המבט של המפתחים וקהל היעד על המערכת ועל תפקודה.

2.1.4. הנחות ואילוצי מערכת

- אדם אשר מתקין את התוכנה יש צורך לדעת אילו הגדרות רשת אמורים להיות לאותו מחשב בו מתקין את התוכנה.
- התוכנה תהיה בשפה אנגלית.

2.2. תיאור כללי

2.2.1. תכלית הפרויקט

המטרה העיקרית של הפרויקט היא לזהות שהמחשב התחבר לרשת זרה.
מי שמתקין את התוכנה יגדיר עבור המחשב אילו הגדרות תקינות לרשת בה אמור להימצא במחשב, ומה התוכנה אמורה לעשות במידה והמחשב מתחבר לרשת זרה כגון נעילת כרטיס רשת, הקלטת תעבורת מחשב, יציאה מהמשתמש, שליחת מייל לגורם רלוונטי, וכדומה.
לאחר התקנת התוכנה, משתמש רגיל של המחשב לא ירגיש בקיום התוכנה אלה אם יתחבר לרשת זרה.

2.2.2. מאפייני התוכנה

2.2.2.1. Task scheduler

שירות של מערכת ההפעלה ווינדוס אשר מאפשרת לאתר אירועי שינוי מצב הרשת והפעלת התוכנה.

2.2.2.2. התקנת תוכנה

בהתקנת התוכנה, מתווספת משימה עבור Task Scheduler אשר תפעיל את התוכנה שלנו לשם ביצוע בדיקות.

2.2.2.3. ממשק משתמש גרפי

- הגדרת נתוני רשת עבור בדיקות.
- הגדרת סדרת בדיקות של רשת.
- הגדרת פעולה לביצוע בעת כישלון של בדיקות רשת.

2.2.2.4. בדיקת ברשת

באחת הבדיקות משתמשים במודל שרת-לקוח לאימות בדיקת רשת. מחשבים באותה רשת אמורים להיות בעלי אותו צופן.

- **צד לקוח:**
  - איסוף מידע על מחשבים ברשת.
  - בדיקת המידע מול הצופן של הלקוח.
- **צד שרת:**
  - שליחת הצופן האישי לבקשת הלקוח.

2.2.3. סביבת עבודה

סביבת עבודה לשימוש:

- מערכת הפעלה Windows XP ומעלה.

סביבת עבודה לפיתוח:

- מערכת הפעלה Windows XP ומעלה.
- Eclipse מומלץ בגרסת Luna.
- Java SE 1.7 ומעלה.

2.3. יעדים ומטרות

מערכת אותה אנו הולכים לייצר עבור יחידה צבאית ממר"ם נועדה להגן על המידע הסודי השמור על המחשב הצבאי. המערכת צריכה לזהות שהמחשב התחבר לרשת זרה, ואם הדבר קרה אז מערכת זו תבצע את המטרה העיקרית ע"י ניתוק המחשב מהרשת הנוכחית והלא תקינה, כגון כיבוי מחשב, יציאה ממשתמש וכדומה. כמו כן מדווחת על האירוע ושומרת נתוני רשת לשימוש עתידי.

2.3.1. יעדים

- יצירת תוכנת עזר שתיתן מענה לסעיפים הבאים:
  - ידידותית למשתמש.
  - אמינה.
  - יעילה מבחינת משאבים
- קבלת נתוני רשת וניתוחם.

2.3.2. מטרות

- בניית תוכנה אמינה וידידותית למשתמש המאתחל את נתוני התוכנה.
- שיתוף פעולה עם צוות ממר"ם.
- יצירת גמישות לשיפור עתידי של התוכנה.

2.4. בעיות

- חלק מהבדיקות תלויות בגישות ברשת כגון:
  - בדיקת אתר פנימי – במידה והאתר לא זמין לא מוכיח שהמחשב מנותק מהרשת התקינה.
  - בדיקת מחשבים אחרים ברשת – לא תמיד זמינה ותלויה בסביבה.
  - כל תקלה ברשת עלולה להוביל לנתונים שגויים ופעולות לא רצויות של התוכנה.
  - התוכנה מתאימה לארגונים ולא למחשבים פרטיים.
  - שליחת מייל – במידה ולא נמצאים באותה הרשת של שרת המייל, התוכנה לא תצליח לשלוח מייל התרעה.

2.5. יישום

2.5.1. סקירת מצב קיים

כיום יש שיטות שונות להגנת מידע הנמצא ברשת ארגונית כגון:

- Fire Walls
- Anti-Virus
- מערכות IDS.

אם זאת שיטות אלו לא עוסקות בזיהוי חיבור לרשת זרה.

2.5.2. אופי המערכת וסוגה

התוכנת שאותה אנו מתכננים הינה חדשה ללא כל זיקה לתוכנה קיימת היום.
התוכנה נועדה לעזור לארגון:

- לזהות חיבור לרשת זרה.
- לשמור נתונים על הרשת הזרה.
- לדווח לגורמים רלוונטיים לגבי חיבור לרשת זרה.
- לבצע פעולות נדרשות על מנת למנוע דליפת מידע בעזרת סוגי פעולות שונות המנתקות את המחשב מהרשת.

2.5.3. אילוצי המערכת

- אילוצי חומרה:
  - אין אילוצי חומרה מיוחדים, רק צריך כרטיס רשת.
- אילוצי תוכנה:
  - התוכנה מותאמת למערכות הפעלה של Windows.
  - המערכת תכלול Java VM 1.7+.
  - המערכת צריכה לאפשר להריץ את התוכנה בהרשאות מנהל.
  - המערכת אמורה לכלול Task Scheduler.

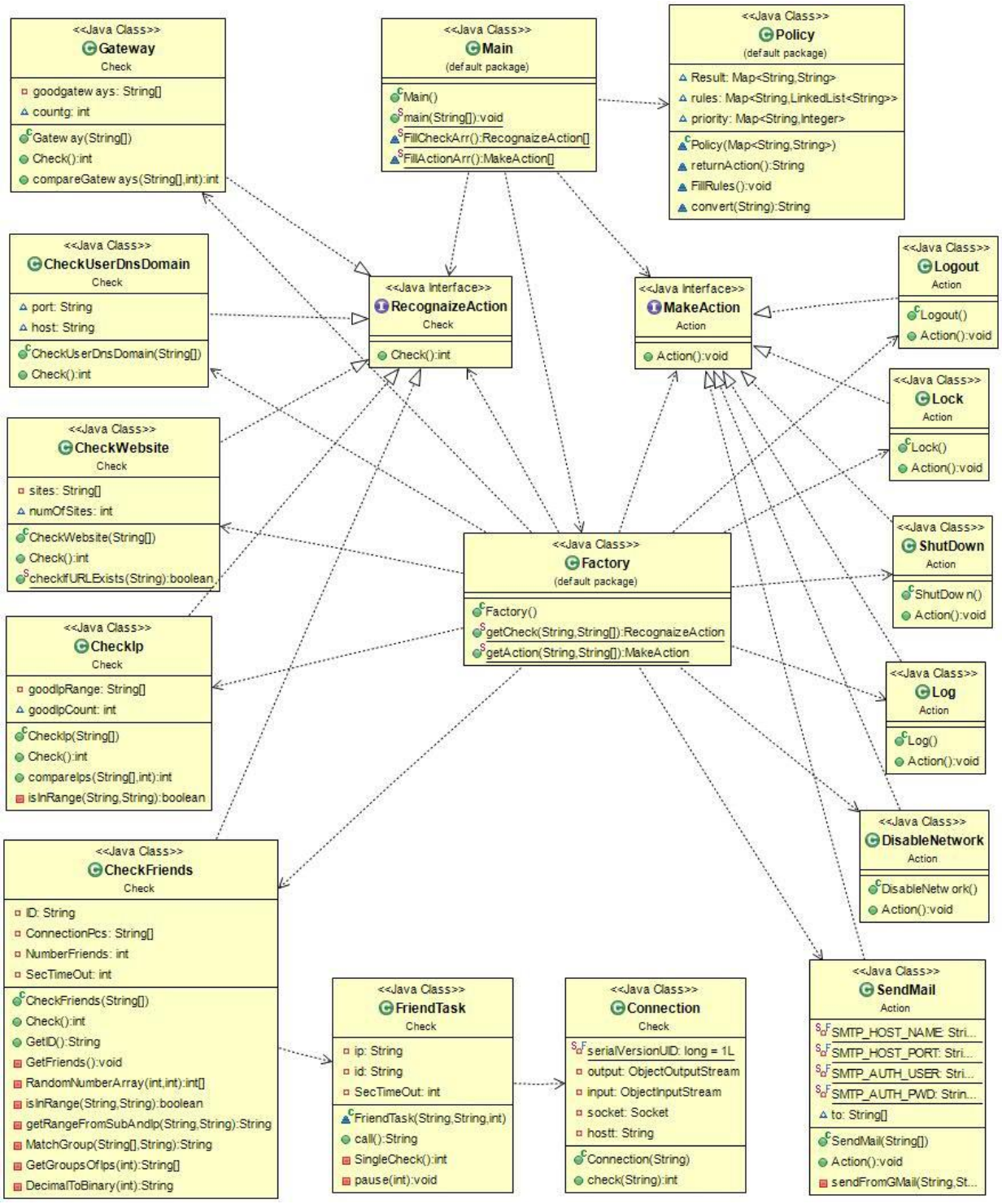
2.5.4. מילון מונחים

IP של המחשב	כתובת IP היא מספר המשמש לזיהוי נקודות קצה, כגון מחשב ,ברשתות תקשורת שבהן משתמשים בפרוטוקול התקשורת IP ,כגון רשת האינטרנט.כתובת זו ניתן להגדיר באופן ידני, או בעת התחברות לרשת ניתן לקבל אותה בצורה אוטומטית.
Default Gateway	אחת מהגדרות הרשת שבזכות הגדרה זו, יודע המחשב מהי הכתובת של ציוד הרשת, שמאפשר תקשורת לכיוון כלל הרשת. לדוגמה, הגדרת שער ברירת מחדל במחשב שנמצא ברשת מקומית ,תאפשר למחשב למצוא את הנתב שמחבר בין הרשת המקומית לבין כלל האינטרנט.
בדיקת Ping	Ping היא חבילת נתונים בפרוטוקול ICMP הנשלחת ממקור מסוים ליעד מסוים ברשת לפי כתובתו. המטרה העיקרית לה היא משמשת היא בחינת תקינות התקשורת בין נקודת המקור לנקודת היעד.
Domain Name	שם תחום או שם מתחם ,הוא שם ייחודי של אתר ברשת האינטרנט, שמבדיל אותו משאר האתרים הנמצאים ברשת. דוגמה :לשם התחום של האתר של עיתון הארץ www.haaretz.co.il il מציין את המדינה שבה נרשם שם התחום. co מציין את המגזר בתוך המדינה. haaretz הוא שמו הפרטי של האתר, במסגרת המגזר המסחרי בישראל. www מציין את כלל האינטרנט
DNS- Domain Name Service	הוא פרוטוקול המאפשר גישה לבסיס נתונים מבוזר ותפקידו להמיר בין שמות לבין מספרי IP וחוסך לנו את הצורך לזכור מספרי IP של אתרי האינטרנט. כאשר אנו מקישים בדפדפן שם של אתר, המחשב פונה לשרת ה DNS-שהוקצה לו ע"י ספק האינטרנט ומקבל בתשובה את מספר ה IP-של אותו אתר וכך הוא יכול לשלוף ממנו את דפי האינטרנט.
מחשב אשר מחובר לDomain	זוהי שיטת עבודה בה כל המחשבים ברשת מחוברים לשרת. כל ההגדרות של המשתמש והנתונים שמורים בשרת. כאשר משתמש רוצה להתחבר למחשב, הוא צריך לבצע הזדהות מול השרת, במידה והצליח יוכל להתחבר למחשב "שלו" , גם אם יום למחרת ינסה להתחבר למחשב אחר , אחרי הזדהות הוא יתחבר שוב למחשב "שלו". בשרת ניתן להגדיר הגדרות על כל המשתמשים , כגון פעולות שאסור למשתמש לעשות, רקע, קישורים, קבצים של המשתמש וכדומה.
Cmd-Command prompt	מעטפת מערכת ההפעלה המבוססת על ממשק שורת פקודה. במערכות הפעלה של Windows ניתן להיכנס ע"י : התחל -> הפעלה-> "cmd"
Task Scheduler	מתזמן משימות הוא רכיב של Microsoft Windows המספק את היכולת לתזמן הפעלה של תוכנות או סקריפטים בזמנים מוגדרים מראש או לאחר מרווחי זמן קבועים.
Daemon	Daemon (דימון) היא תוכנית מחשב שרצה כתהליך רקע, להבדיל מתוכניות הנמצאות תחת שליטתו הישירה של משתמש אינטראקטיבי. בדרך כלל מערכות מפעילות Daemon בזמן האתחול, ולרוב הם משרתים פונקציות כגון תגובה לבקשות רשת, לפעילות חומרה, או לתוכניות אחרות על ידי ביצוע של משימה כלשהי.
רשת זרה	רשת זרה היא כל רשת שהמחשב יכול להתחבר אליה חוץ מהרשת הארגונית.
System Call	היא בקשה של תוכנת מחשב ממערכת ההפעלה לבצע פעולה שהיא אינה יכולה לבצע בעצמה. קריאות מערכת משמשות את התוכנות לגישה למרבית רכיבי החומרה של המחשב (למשל קריאת קובץ מהדיסק הקשיח) ,ליצירת תהליך חדש, להעברת מידע בין תהליכים ועוד.

2.5.5. ממשק תפעולי

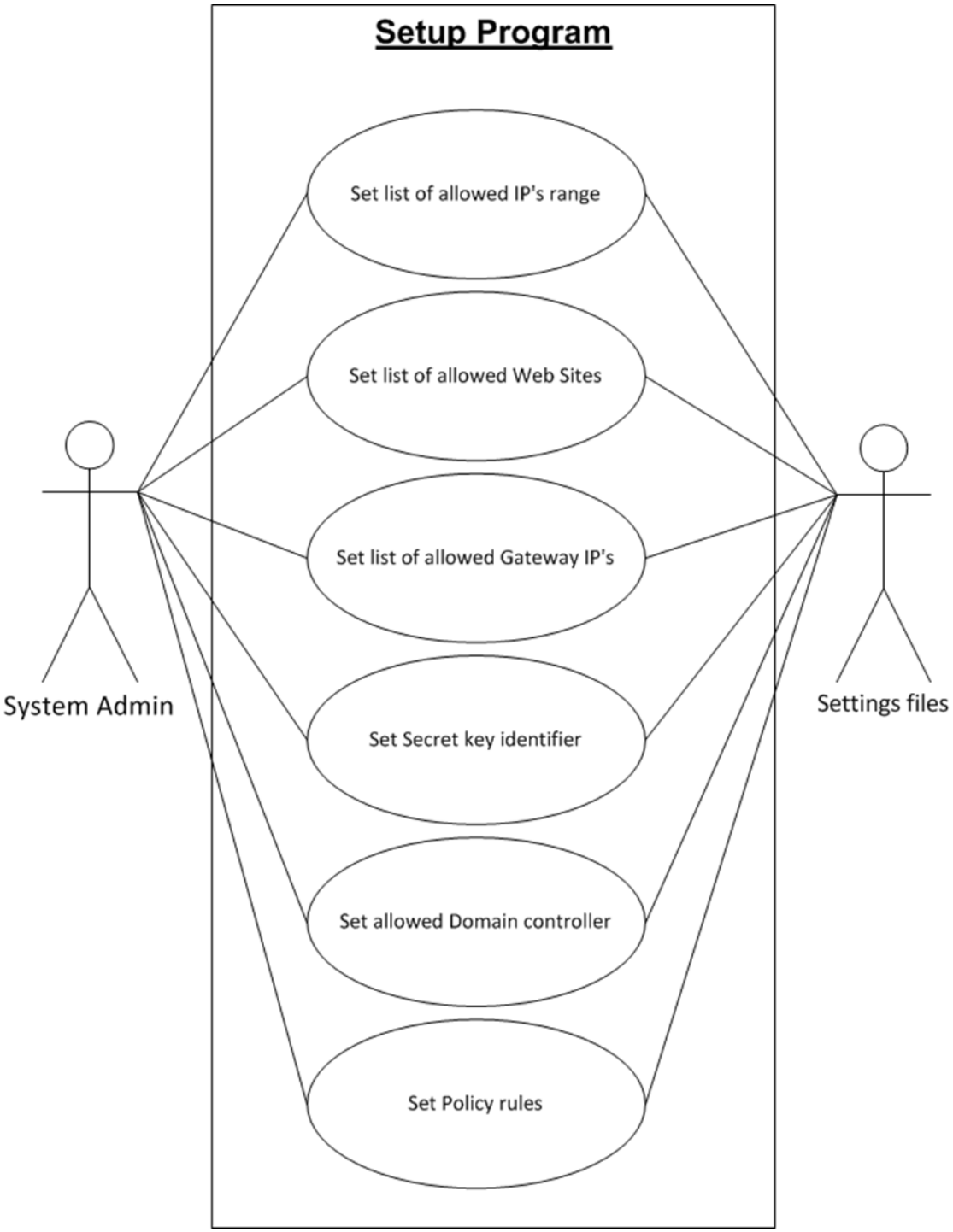
משתמשי המערכת חייבים להיות בעלי השכלה בתחום הרשת הארגונית על מנת להגדיר הגדרות לתוכנה לשם עבודתה. המערכת תעבוד על פלטפורמת JVM על מערכות Microsoft Windows.

2.5.6. Class Diagram

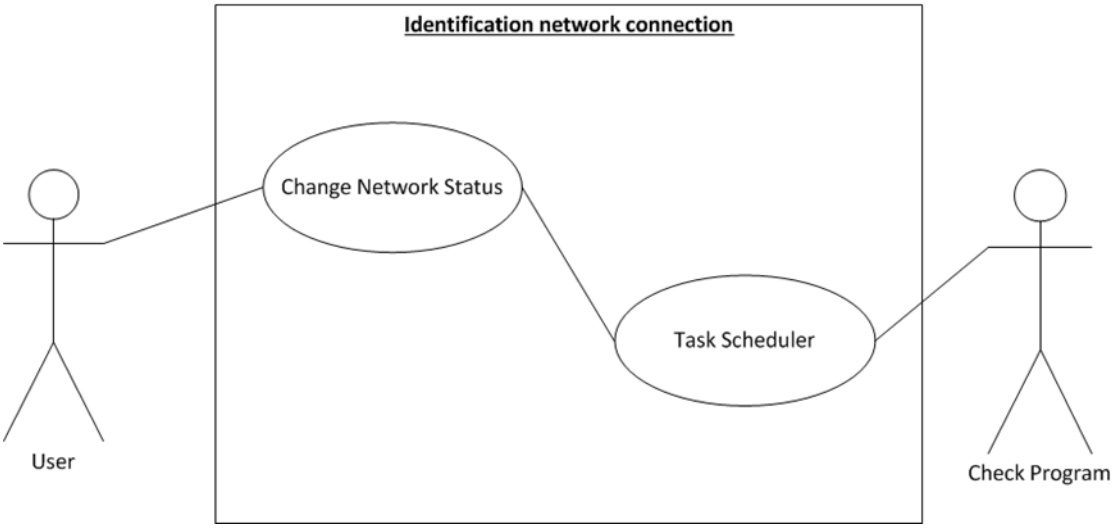


2.5.7. סקירת ה– Use Case

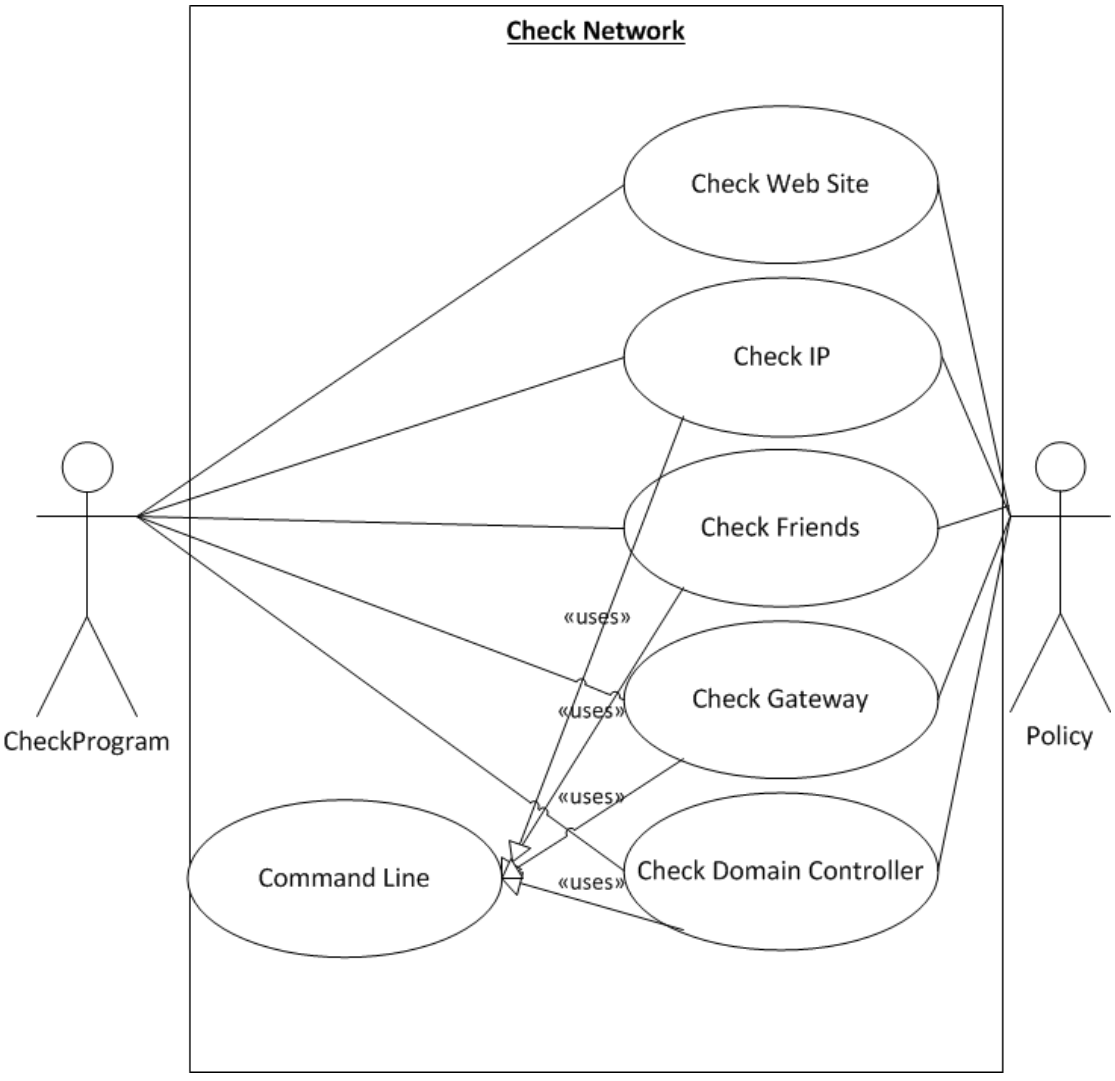
2.5.7.1. הגדרת התוכנה



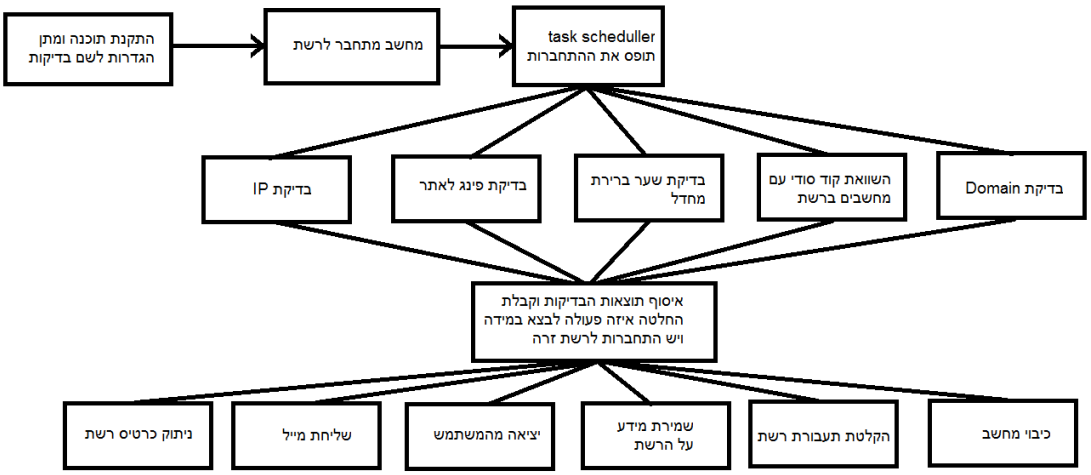
2.5.7.2 זיהוי התחברות לרשת



2.5.7.3 בדיקת רשת



2.5.8.מבנה המערכת



2.5.9. מאפייני המערכת

<u>דרישות פונקציונליות</u>:

- תחום העבודה :
  - מנחה פרויקט: אחראי ללוות ולעקוב אחר התפתחות הפרויקט מתחילתו ועד סופו.
  - מפתחים אחראים לפיתוח מוצר התוכנה:
    - ❖ פיתוח ממשק נוח לשמירת ההגדרות.
    - ❖ פיתוח התוכנה בצורה כך שהתוכנה תהייה גמישה לשינויים ולהוספות.
- תחום המוצר:
  - על המנהל רשת להתקין את התוכנה על כל מחשב אשר מחובר לרשת.
  - בכניסה הראשונה למערכת המנהל רשת צריך להגדיר למערכת אילו בדיקות יבוצעו ולתת הגדרות לתוכנה.
  - לאחר התקנת התוכנה , משתמש רגיל לא חש שהמערכת קיימת.
  - בעת התחברות לרשת, התוכנה מתחילה לעשות בדיקות לפי ההגדרות כך שהמשתמש לא מרגיש שזה נעשה.
  - במידה ויש התחברות לרשת זרה, התוכנה תפעל בהתאם למה שקבע מנהל הרשת.

<u>דרישות לא פונקציונאליות</u>:

- מתן הגדרות לתוכנה:
  - עיצוב נוח להכנסת כל הנתונים הנדרשים עבור התוכנה.
  - אופציה לשנות הגדרות גם אחרי הכניסה הראשונה של מנהל הרשת.
- התוכנה תהיה בעלת התכונות הבאות:
  - ניתנת להרחבה בקלות - קוד מאורגן כך שניתן להוסיף בדיקות חדשות, פעולות חדשות ושיוך הדברים לתוכנה הכוללת.
  - ניתנת לתחזוקה - התוכנה כתובה בצורה ברורה לפי שיטות העיצוב המקובלות כך שבעתיד יהיה ניתן לשדרג אותה במידת הצורך או לשכתב בשפה אחרת במידת הצורך.
  - נוחה לשימוש - התוכנה תמומש בצורה כך שמתקין התוכנה ידע בקלות איך להגדיר את כל ההגדרות.
  - אמינות - המערכת תבצע את הנדרש עליה בדיוק לפי הגדרות המערכת.
  - חוסכת במשאבי המערכת - המערכת תפעיל את הבדיקות פעם אחת אך ורק כאשר יש התחברות לרשת חדשה.

2.6. טכנולוגיה ותשתית

הארכיטקטורה מבוססת על שכבת יישום :

- Java IDE (Integrated Development Environment) - זוהי סביבת פיתוח משולבת המסייעת למתכנתים לפתח תוכנה. היא כוללת לרוב עורך קוד מקור, מהדר או/ו מפרש, כלי בניה ממוכנים ו - Debugger. הסביבה כוללת כלים לבקרת תצורה וניהול גרסאות וכן כלים המקלים על בניית יישומים בעלי ממשק משתמש גרפי. סביבות מודרניות כוללות גם כלים המסייעים בתכנות מונחה עצמים. היא מקלה על ההליך התכנותי ,והופכת אותו לפחות מסורבל.
- Batch Programming Language – זוהי שפה המכילה שורות טקסט של פקודות למערכת ההפעלה, לשם הפעלתן של תוכנות שונות. בדרך כלל רושמים בתוך קובץ עם סיומת bat. בפיתוח תוכנה בשפת Java משתמשים בפקודות אלו על מנת לבצע System Call אשר מאפשר לנו גישה לחומרה של המחשב.

3. ניהול עבודה בצוות

כאשר התחלנו לעבוד על פרויקט, נאלצנו להתמודד עם בעיות וקשיים הנוגעים לארגון העבודה בצוות, וחלוקה נכונה של משימות. מכאן, נבעו הבעיות הבאות שדרשו פתרון.

- חלוקת משימות - מהן המשימות ? מי עובד על כל משימה ? מהו סדר ביצוע המשימות ?
- זמנים ותאריכים - כמה זמן יש להקדיש לכל משימה כך שנסיים את הפרויקט בזמן?
- ארגון וניהול כתיבת הקוד - איך משלבים קוד של כמה אנשים? איך מתממשקים בין המחלקות?

כמענה לשתי הבעיות הראשונות, בחרנו להשתמש במתודולוגיית פיתוח תוכנה זריז (Agile). עבור ארגון נכון של כתיבת הקוד . בכל שבוע \ שבועיים נערך מפגש שבו דנו בהתקדמות הפרויקט ובהמשך המשימות לשבוע הקרוב.

3.1. מתודולוגיית פיתוח תוכנה

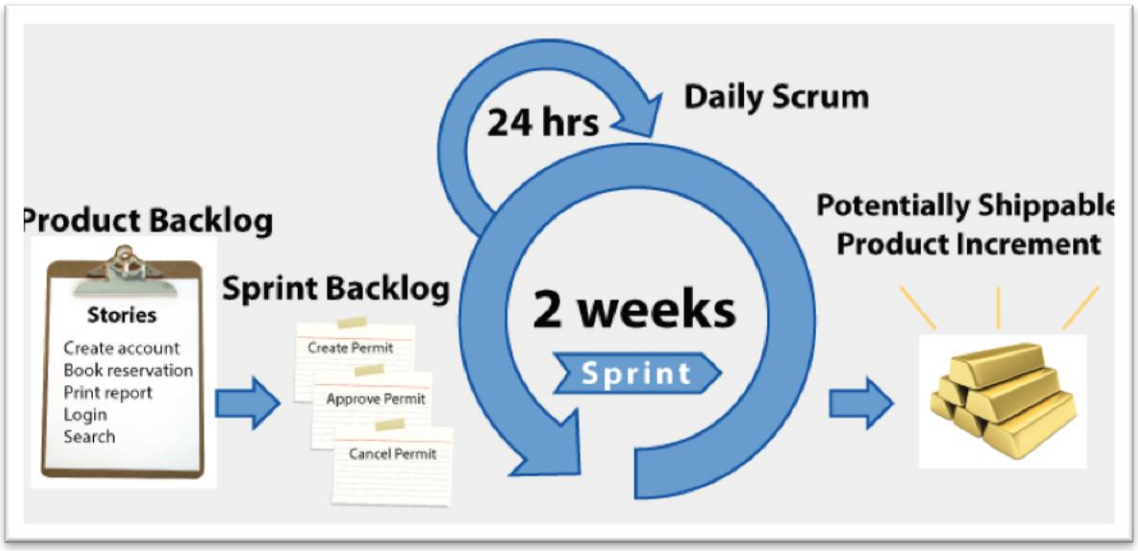
בהנדסת תוכנה, מתודולוגיית פיתוח תוכנה היא סט מוסכם של עקרונות, תהליכים, פעילויות וכלים על פיהם מפותחות ומתוחזקות מערכות תוכנה. יש כיום כתריסר מתודולוגיות עיקריות (שפורסמו ברבים וזכו לקבלה מסוימת בתעשייה), וכן מאות אחרות, משניות. המתודולוגיות נבדלות ביחסן למקצוע הנדסת התוכנה ("מהי הנדסת תוכנה?") ועקרונות היסוד, המיקוד (ניהול פרויקט, ארכיטקטורה, עיצוב ,תכנות, והבטחת איכות), הטכניקות ובהיבטים נוספים. בשל גילו הצעיר של הענף ובשל ריבוי המתודולוגיות אין עדיין הסכמה באשר למידת התאמתן של מתודולוגיות מסוימות לבעיות מסוימת, אם כי מקובל לחלק את המתודולוגיות למשפחות נפרדות.

3.1.1. פיתוח תוכנה זריז (Agile)

התפיסה האג'ילית לפיתוח תוכנה מבוססת על הנחת העבודה כי תהליכי פיתוח תוכנה מאופיינים בשינויים רבים, ולכן, יש לבנות עבורם מנגנון ניהול המתמודד בהצלחה עם מאפיין זה. זוהי מתודולוגיה איטרטיבית שהותאמה לפיתוח תוכנה בצוותים קטנים תוך שימת דגש על יעילות, זריזות ואיכות. הדגש על פיתוח תוכנה איכותית מתבטא הן מבחינת קיום דרישות הלקוחות והן מבחינת העדר באגים. רעיונותיה של התפיסה האג'ילית מיושמים באמצעות מספר מתודולוגיות פיתוח תוכנה אג'יליות, כמו למשל, Extreme Programming Lean Software Development, Crystal, Scrum.

3.1.2. SCRUM

Scrum היא מתודולוגיה זריזה לניהול פרויקטים לפיתוח תוכנה. המתודולוגיה פותחה באמצע שנות ה- 90 על ידי קן שוואבר וג'ף סאתרלנד. השיטה מתבססת על ההנחה שפיתוח תוכנה הוא בעיה אמפירית ולא ניתן לפתור אותה בשיטות מסורתיות המתבססות על חיזוי או תכנון. Scrum מניחה שלא ניתן להבין או להגדיר פיתוח תוכנה מסוימת במלואה ומראש, ובמקום זאת מתמקדת בשיפור יכולתו של הצוות לספק תוצרים במהירות ולהגיב לדרישות העולות תוך כדי התהליך. כמו כן, השיטה שמה דגש על צוותים בהכוונה עצמית, המנווטים את הפיתוח באופן עצמאי.



תהליך פיתוח תוכנה בשיטת ה - Scrum כולל כמה שלבים עיקריים:

- Product Backlog - תחזוקה של רשימת פריטי העבודה לביצוע, מסודרים לפי קדימויות.
- לוח ספרינטים השלמת מנה קבועה של פריטי עבודה בסדרה של איטרציות קצרות המכונותSprint . בסיום הספרינט, המאפיינים שהוגדרו ב - Product Backlog מקודדים, נבדקים ואז משתלבים במערכת המתפתחת.
- פגישת צוות פעם בשבוע (המכונה ' Stand up Meeting '). בפגישה מציג כל אחד מחברי הצוות את ההתקדמות, העבודה המתוכננת וקשיים אפשריים.
- פגישת תכנון ספרינט (Sprint Planning) קצרה שבה מוגדרים פריטי העבודה לאותו Sprint.
- פגישת ניתוח ספרינט (Sprint Retrospective) קצרה להפקת לקחים מה - Sprint הקודם.
- בסוף של כל ספרינט, מתבצעת סקירה של הספרינט, שבמהלכה הצוות מדגים את הפונקציונאליות החדשה, אשר מספק משוב, וזה יכול להשפיע על הספרינט הבא.

3.2. תכנית עבודה בפרויקט

שלבים לתכנון ביצוע משימות להשלמת הפרויקט, כל ספרינט מתבצע בין שבוע לשבועיים ובכל סיום ספרינט מתבצעת ישיבה לאינטגרציה של משימות הספרינט

<u>ספרינט 1:</u>

- חקר של מאפייני רשת ארגונית .
- מימוש אלגוריתם לעבודה עם קבצי הגדרות.
- מימוש מודל לקוח עבור בדיקה של "חברים ברשת" (אשר ממש ממשק RecognizeAction ששולח מסר לחברים ברשת לוקאלית) .
- לעשות פרויקט חדש שזהו השרת עבור בדיקה "חברים ברשת" .
- מימוש מחלקות שממממשות ממשק MakeAction : כיבוי מחשב , Logout , Lock , לשלוח מייל, ניתוק מהרשת , בדיקת IP, בדיקת PING.

ספרינט 2:

- לתקן מודל רשת/לקוח לבדיקה "חברים ברשת"
  - להוסיף מצב בו הלקוח מנסה 3 פעמים להתחבר לשרת על מנת לקבל תשובה עם זמן השהייה בין ניסיונות החיבור.
  - לבצע בדיקה עבור מספר חברים ברשת ואם לפחות אחד הצליח לעבור את הבדיקה אז הבדיקה בסדר.
  - בדיקת חברים ברשת דרך CMD.
- מימוש Factory Method - שיוצר אובייקט מתאים בעזרת שם של אובייקט ומערך של פרמטרים (כל מחלקה תדאג לפרק פרמטרים אלו בעצמה).
- פונקציה ב -main אשר קוראת מהקובץ בדיקות ובעזרת Factory Method יוצרת מערך של אובייקטים של בדיקות שיש לבצע, כמו כן ,מקובץ פעולות יוצרת מערך של אובייקטים עם כל הפעולות.
- לבצע LOG של הרשת ולשמור בו נתונים על הרשת.
- כתיבה לקובץ – מחלקה שכותבת לקובץ הגדרות.

ספרינט 3:

- מימוש Thread Pool - על מנת לבצע בדיקות של "חברים ברשת" במקביל- אם לפחות אחד החזיר "אמת" אז הבדיקה עוברת.
- מימוש Class Policy וביצוע אינטגרציה לתוכנית ראשית.
- מימוש ממשק משתמש GUI – עבור כתיבה וקריאה מקובץ.

ספרינט 4:

- תיקון מחלקה Checkfriend -פתרון בעיה של הריגת Threads שנימצא ב- Thread Pool.
- עבור מחלקה Policy - יש לשפר את Action שמוחזר. (לא לשכוח למיין את המערך לפי עדיפויות של קובץ Policy. שורה עליונה עדיפות הכי גבוהה).
- תיקון ממשק משתמש GUI
  - להוסיף אפשרות Disable/Enable לChecks , כלומר במידה ו Check לא מסומן , כל הדברים הרלוונטיים נהיים Disable.
  - לתת אופציה להוסיף action מסוים עם קומבינציות שונות של בדיקות.
  - קבצים של read-write - ליישם.
- להוסיף Checks נוספים כגון בדיקת Domain .
- לשפר קובץ LOG שנישמר.
- לארגן את הקוד ב - Package כלומר כל Actions ב - Package אחד וכל ה-Checks ב - Package אחר.
- לעשות בדיקה מקיפה של כל ה - Checks, במידה ונכשלו החזרה "1", ובמידה ועברו מחזירה "0".
- להתחיל לכתוב בדיקות יחידה Java Unit Tests .

ספרינט 5:

- לתקן LOG - שם של הקובץ צריך להיות תאריך ושעה.
- לתקן Domain server כך שתהיה בדיקה האם הוא חי.
- לתקן את הפעלת LOG כך שיתבצע תמיד במידה ונכשלה אחת מהאופציות של Policy.
- תיקון ממשק משתמש GUI
  - רישום לקובץ Policy - ניתן להזין Action ספציפי רק פעם אחת.
  - להוסיף Domain name

ספרינט 6:

- לבצע תיקונים אחרונים ולייצר קבצי הרצה.
- לבצע בדיקות מערכת:
  - לעשות בדיקות לכל ה- Checks.
  - לעשות בדיקות לכל ה- Action.
  - לעשות בדיקות שהתוכנה מותקנת כראוי.
  - לעשות בדיקות לממשק GUI.
  - לעשות בדיקות לשרת שמשמש לבדיקה של "חברים ברשת".

4. שפת הפיתוח : Java

התוכנית פותחה בסביבת העבודה Eclipse IDE.

4.1. Java Language

המטרה העיקרית שעמדה לנגד עיניהם של מפתחי השפה הייתה ליצור שפה אשר מאפשרת לכתוב את התכנית פעם אחת ולאחר מכן להריץ אותה בכל מחשב מבלי לבצע שינויים. קובץ קוד המקור של תכנית ב - Java מתורגם ל- Java Byte Code שנשמר בקובץ ששמו זהה (בדרך כלל) לשמו של קובץ קוד המקור, והסיומת שלו ".class" קובץ ה - class ניתן להרצה בכל מחשב הודות ל - JVM. ה - JVM היא תכנית שנכתבה במיוחד למערכת הפעלה מסוימת (קיימות גרסאות של ה - JVM כמעט לכל מערכת הפעלה קיימת). תכנית זו מסוגלת לתרגם את קובץ ה - Java Byte Code (הקובץ עם הסיומת .class) לשפת המכונה של המחשב שבו היא פועלת. בדרך זו מושגת אחידות שלא הייתה קיימת בשפות אחרות. בדרך זו ל - Java יש רמת תאימות גבוהה לא רק בכל הקשור ל - Source Code אלא גם בכל הקשור לקובץ שמוכן להרצה.

ב - Java קיימות מחלקות מוכנות אשר כוללות מתודות/פונקציות שמקלות על כתיבתן של תכניות אשר מתקשרות עם תכניות אחרות (במחשבים שונים). המחלקות המוכנות ב- Java כוללות תמיכה במגוון רחב של פרוטוקולים (HTTP, TCP\IP, UDP, SMTP ואחרים).

אחת הפעולות שמעיקות על כל מתכנת הוא כתיבת התיעוד לתכנית שכתב. מפתחי השפה חשבו על בעיה זו מראש, ופיתחו כחלק מהשפה את ה - Javadoc . ה - Javadoc היא תכנית שבעת הפעלתה על תכנית הכתובה ב - Java היא יוצרת דפי HTML אשר מכילים תיעוד מפורט לתכנית עפ"י ההערות המתאימות שנשתלות בקוד המקור.

ה- API שמשמש כ - help לכל מפתח ב - Java נוצר באמצעות הפעלת ה - Javadoc על קבצי קוד המקור של המחלקות שיש ב- Java.

אופן כתיבתן של תכניות אשר חלקים שונים מתוכן מתבצעים במחשבים נפרדים תוך כדי העברת אינפורמציה בינם לבין עצמם פשוט בהרבה מאשר בשפות תכנות אחרות.

4.1.1. Java Virtual Machine

ה - JVM – Java Virtual Machine - אחראית לתרגום ה - Java Byte Code לשפת המכונה שעליה התכנית מורצת .ה - JVM יכולה להיות תוכנית שמורצת במחשב או אפילו חלק מהחומרה שלו. קובץ ה- Java Byte Code (הקובץ עם הסיומת .class) מכיל למעשה סדרת הוראות ל - JVM. בכל כלי שמסוגל להריץ תכניות ב- Java קיים ה-JVM. ה JVM Specification -מכיל הגדרות מדויקות בנוגע לאופן שבו אמורים להיות קבצי ה - Java Byte Code כדי שיוכלו לרוץ באמצעות ה - JVM .כאשר מגיעה מחלקה מרשת האינטרנט לפני שהיא מורצת מופעל ה - Byte Code Verifier אשר מוודא בין היתר את עמידת קובץ ה Java - Byte Code בכללים.

4.2. למה בחרנו בשפת תכנות Java

תכנות ב- Java טובה מהסיבות הבאות:

- Java תומכת בכל מערכות ההפעלה והפיתוח שלו לא עולה כסף.
- תומך בתהליכונים בצורה קלה עבור אפליקציות צד שרת. כאשר לכל חיבור לקוח או משימה צריך ליצור תהליכון משלו.
- סביבת עבודה ב- Java היא בטוחה, וכתיבת התוכנה חסינה מנפילת מערכת.
  - החוסר בשימוש במצביעים מבטל את כל סוגי בעיות הזיכרון בעת תהליך התכנות.
  - כל סוגי החריגות וטעויות ניתנות לתפיסה וטיפול, אפילו שגיאה של out-of-memory יכולה להיות לא קריטית.
  - שגיאה שלא נתפסה בזמן ריצה היא קריטית רק לתהליכון שמריץ אותה, ולא לתהליך כולו.
- קיימות הרבה ספריות עם פונקציות מוכנות אשר מקלות על המתכנת, ולא צריך "להמציא גלגל מחדש".

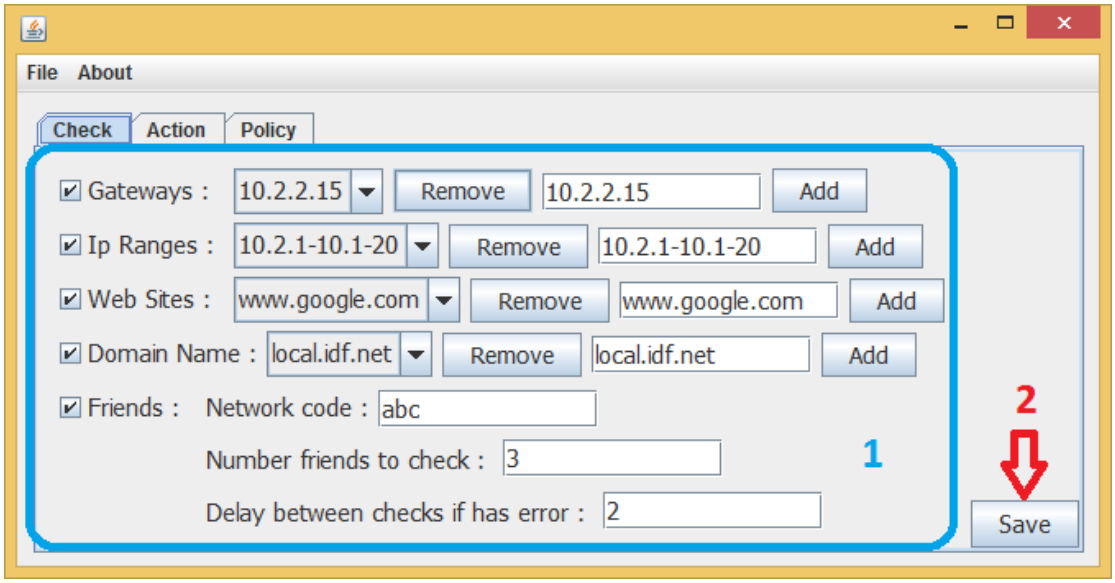
5. UI – (User Interface)

בעבר , כאשר מהירות המחשבים הייתה נמוכה , נהגו להשתמש בממשק משתמש טקסטואלי , כזה המבוסס רק על אותיות , ובנוסף להן כפתורי פעולה במקלדת .
מסוף שנות ה- 80 נפוץ ממשק משתמש גרפי (GUI - Graphical User Interface) שבו משלבים אלמנטים גרפיים כתמונות וצלמיות . הממשקים הגרפיים מאפשרים תצוגה גמישה יותר , במיוחד בצבעים וגופנים , והבהרת כוונת הממשק והשימוש בה , באמצעות סמלים וצלמיות , ועבודה קלה יותר במחשב .
בפרויקט בחרנו לממש ממשק גרפי כדי להקל על המשתמש בהגדרת התוכנה, בגלל שהגדרות התוכנה רשומים בקבצי טקסט ויש תחביר מיוחד בו מסודרים הנתונים בקובץ.
למשתמש אין צורך לזכור את כל האופציות של התוכנה ובאיזה סדר להכניס את הנתונים, על מנת שהתוכנה תעבוד כראוי. כמו כן רוב משתמשי התוכנה לא יהיו מודעים למחלקות השונות של התוכנה ולא יוכלו להגדיר אותה ללא שימוש בממשק גרפי.

5.1. Swing

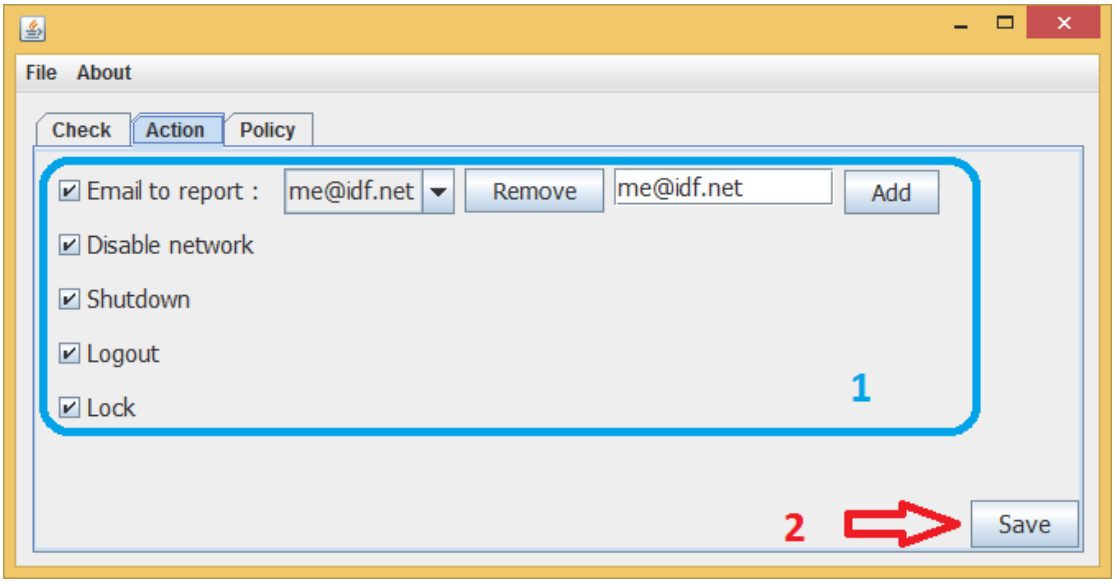
Swing היא ערכת כלים ליישום GUI ב - Java. היא פותחה על מנת לפשט את העבודה ולפתור בעיות שהיו קיימות בטכנולוגיה קודמת שנקראת AWT .
ה - Swing מספק רכיבים מוכנים כמו כפתורים, תיבות גלילה, פאנלים, עצים, רשימות וכו'.

5.1.1. חלון הגדרת Check



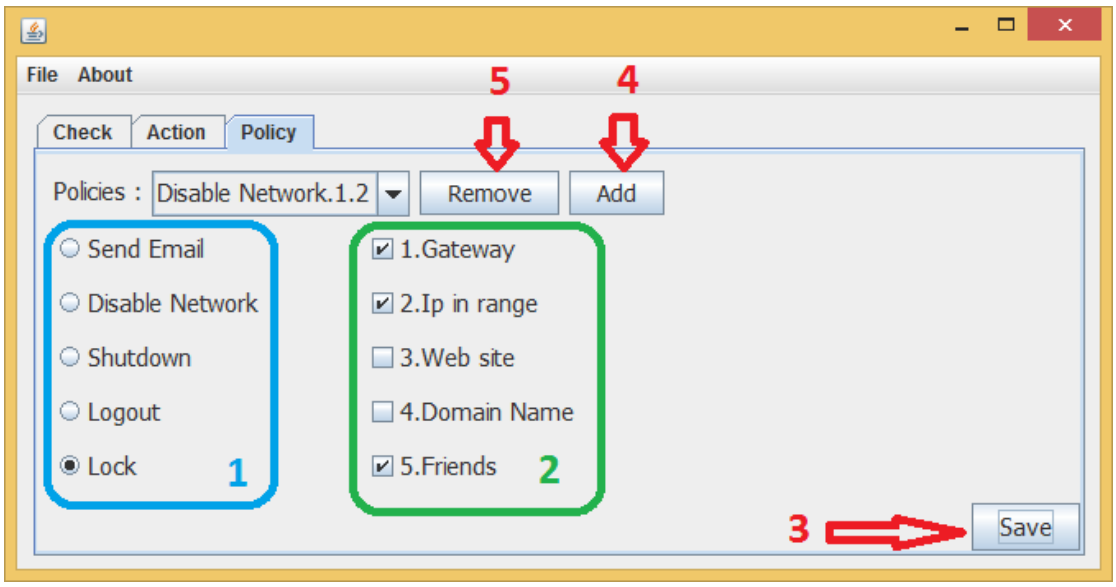
1-בחירת בדיקות לביצוע והגדרות לכל בדיקה.
2-כפתור שמירה ומעבר לחלון הגדרת Action.

5.1.2.חלון הגדרת Action



1-בחירת פעולות לביצוע והגדרות לכל פעולה.
2-כפתור שמירה ומעבר לחלון הגדרת Policy.

5.1.3.חלון הגדרת Policy



1-בחירת פעולה לביצוע .
2- בחירת בדיקות שאמורות להיכשל לביצוע פעולה מסעיף 1.
3- כפתור שמירה.
4- כפתור הוספת חוק.
5- כפתור הסרת חוק.

6. תיאור השיטה

השיטה שאנחנו משתמשים בפרויקט שלנו מתבססת על שרות מובנה במערכת הפעלה Windows שנקרא Task Scheduler ואלגוריתמים לזיהוי רשת זרה.

6.1. Task Scheduler

בעזרתו אנו מבצעים ניטור של חיבור/ניתוק רשת ובעת זיהוי חיבור לרשת חדשה מפעילים אלגוריתמים לביצוע פעולות.

6.2. אלגוריתמים לבדיקת רשת

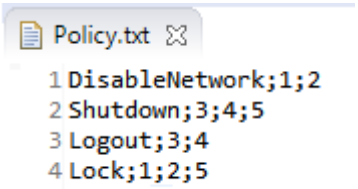
<u>הבדיקה מתבצעת על המאפיינים הבאים</u>:

- בדיקת מחשבים ברשת
  - שליחת מסר למחשבים ברשת לוקאלית
  - קבלת תשובה מהמחשבים שנשלח אליהם המסר
  - שמירת תוצאות בדיקה
- בדיקת IP
  - קבלת כתובת IP של המחשב ע"י פקודת cmd .
  - בדיקת IP שקיבלנו מול רשימה של IP שמוגדרים בקובץ הגדרות.
  - שמירת תוצאת בדיקה.
- בדיקת Domain
  - קבלת שם של Domain ברשת ע"י פקודת cmd .
  - בדיקת של Domain שקיבלנו מול רשימה של Domain שמוגדרים בקובץ הגדרות.
  - שמירת תוצאת בדיקה.
- בדיקת Gateway
  - קבלת כתובת Gateway של הרשת ע"י פקודת cmd .
  - בדיקת Gateway שקיבלנו מול רשימה של Gateway שמוגדרים בקובץ הגדרות.
  - שמירת תוצאת בדיקה.
- בדיקת אתר פנימי
  - שליחת מסר Http Request לאתרים מוגדרים.
  - בדיקה שאתרים מתוך קובץ הגדרות מחזירים לנו תשובה.
  - שמירת תוצאות בדיקה.

אחרי סיום ביצוע של אלגוריתם לבדיקת הרשת מקבלים תוצאות.

6.3. Policy

בשלב ה - Policy טוענים קובץ חוקים שבעזרתו אנו בודקים לאיזה חוק תוצאות של הבדיקה נכשלו ולפי זה מבצעים פעולה שמוגדרת בקובץ Policy .

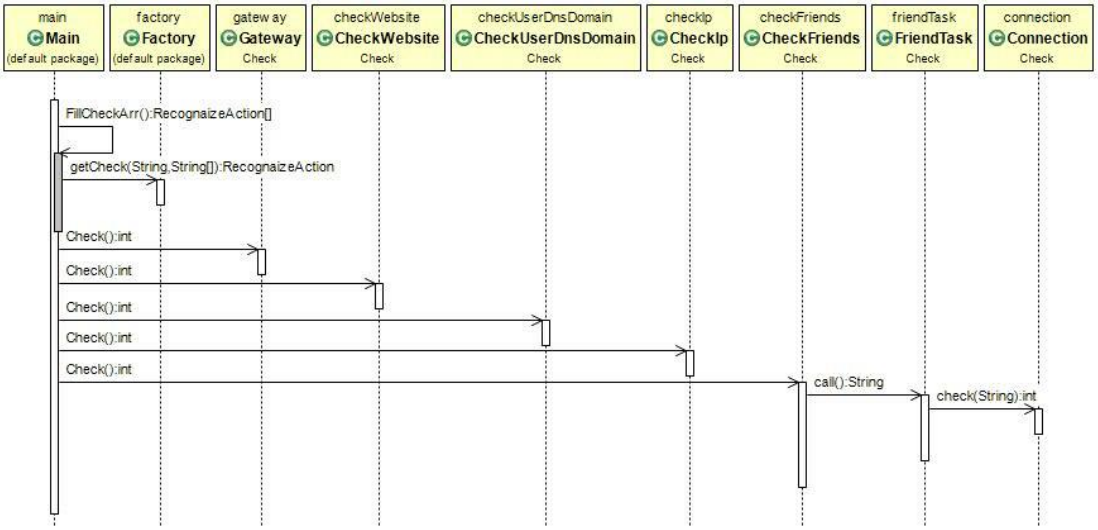


בכל שורה בקובץ מוגדר חוק אחד, בחלק הראשון מוגדר שם הפעולה שצריך לבצע במידה והבדיקות שבאות אחריו לפי קוד של הבדיקה (המופרדות ב";") נכשלות.

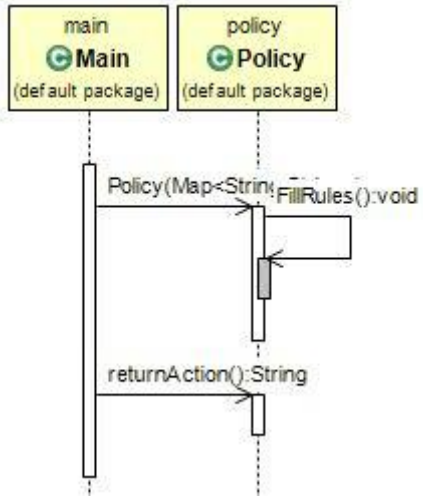
7. UML

7.1 Sequence Diagrams

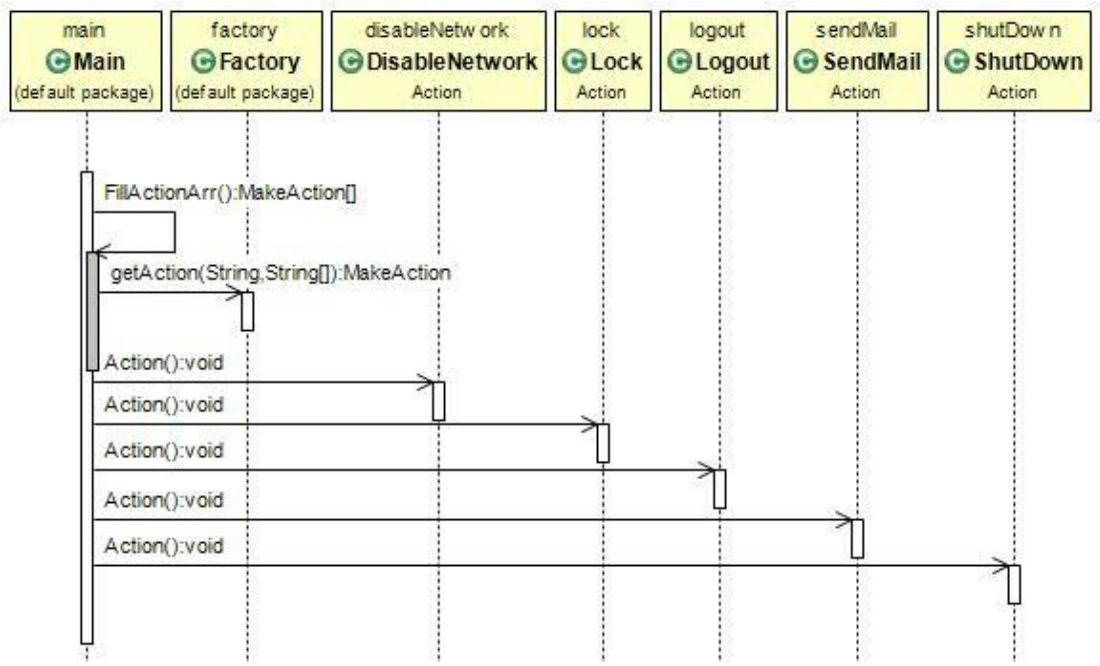
7.1.1 ביצוע בדיקות סביבת רשת



7.1.2 בחירת פעולות לפי Policy



7.1.3.ביצוע פעולות



8. ארכיטקטורת תוכנה

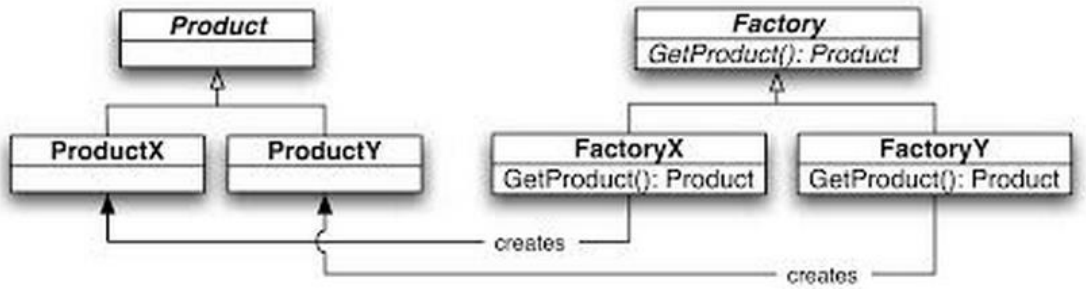
ארכיטקטורה היא התחום העוסק בתכנון מערכות תוכנה . המונח ארכיטקטורה בהנדסת תוכנה פירושו ייצוג היבטים שונים של התוכנה באופן מופשט . ארכיטקטורה של מערכות תוכנה היא לפיכך תכנון מפושט של ההיבטים השונים של התוכנה , היחסים בין המרכיבים השונים של התוכנה והחוקים החלים עליהם.

8.1. תבניות עיצוב (Design Patterns)

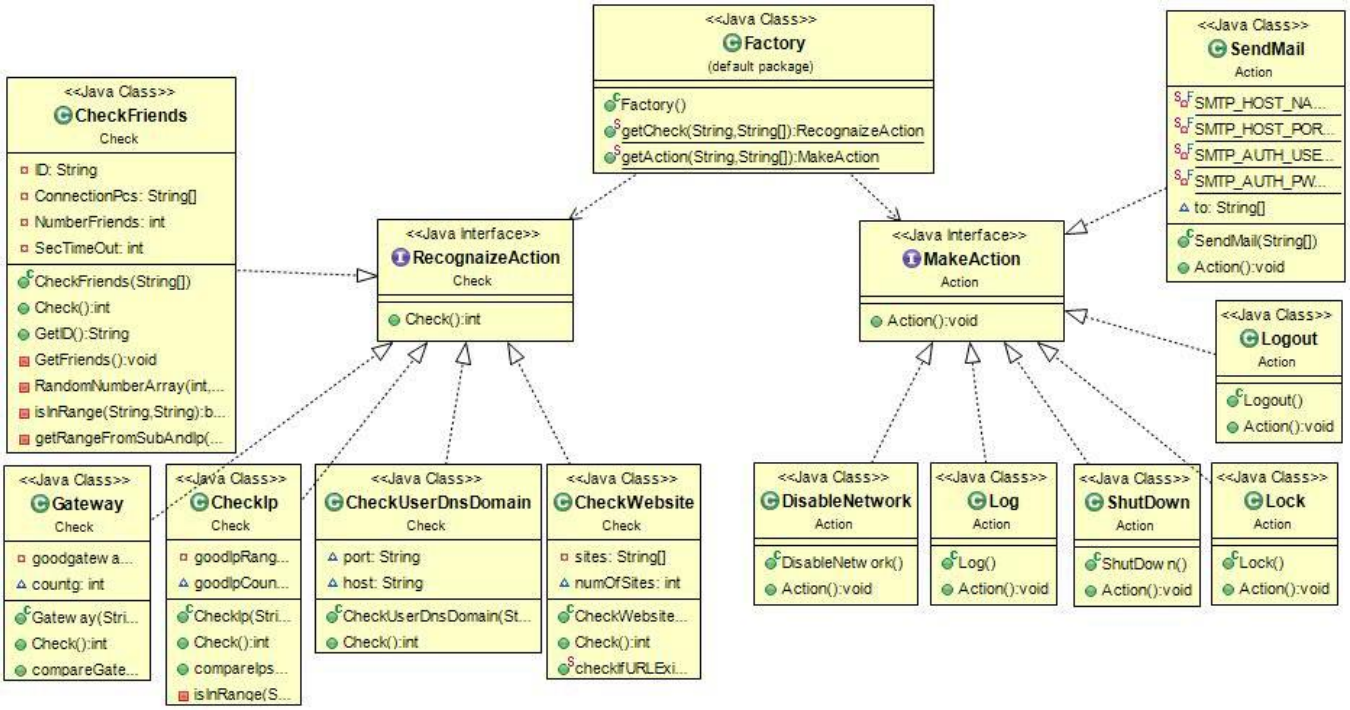
בהנדסת תוכנה , תבנית עיצוב באנגלית : (Design Pattern) היא פתרון כללי לבעיה שכיחה בעיצוב תוכנה . תבנית עיצוב אינה עיצוב סופי שניתן להעבירו הישר לקוד , אלא תיאור או תבנית לדרך לפתרון בעיה, שעשויה להיות שימושית במצבים רבים. תבניות עיצוב מונחות עצמים מציגות לרוב יחסים וקשרי גומלין בין מחלקות או אובייקטים , בלי לפרט את המחלקות או אובייקט היישום הסופיים המעורבים . בפרויקט נעשה שימוש בכמה סוגים של תבניות עיצוב . בפרק זה נפרט על כל סוג ונסביר כיצד השימוש בו בא לידי ביטוי בפרויקט.

8.1.1. תבנית Factory Method

תבנית Factory Method (שיטת המפעל) היא תבנית עיצוב שתכליתה יצירת אובייקטים מבלי להכיר את המחלקות שלהם. התבנית מתבססת על הגדרת שיטה עצמאית ליצירת עצמים. מחלקות שיורשות יכולות לדרוס את השיטה בשיטה משלהן וכך ניתן לציין את הטיפוס המפורש המתבקש.
מהות תבנית המפעל היא להגדיר ממשק ליצירת עצם תוך מתן האפשרות לתת-המחלקות המממשות את הממשק להחליט לאיזה מחלקה ליצור מופע. יצירת המופע נדחית לתת המחלקות. מקובל להשתמש בתבנית ב - Toolkit וב-Framework שם הקוד צריך לייצר עצמים מסוגים שונים אשר עשויים להיות נורשים על ידי אפליקציות אחרות. כן נעשה שימוש בתבנית כאשר עצמים מהיררכיה אחת נדרשים ליצור אובייקטים מתאימים מהיררכיה אחרת. פונקציות מפעל מבצעות כימוס (אנקפסולציה) ליצירת עצמים. הדבר שימושי כאשר תהליך היצירה של מחלקה מורכב ותלוי בהרבה גורמים למשל בגורמי תצורה או קונפיגורציה של האפליקציה או בקלט משתמש.



בחרנו להשתמש בתבנית Factory Method ליצירת אובייקטים של בדיקות ופעולות בהתאם לקובץ הגדרות כאשר זה מאפשר לנו בצורה דינאמית שימוש באובייקט מתאים.

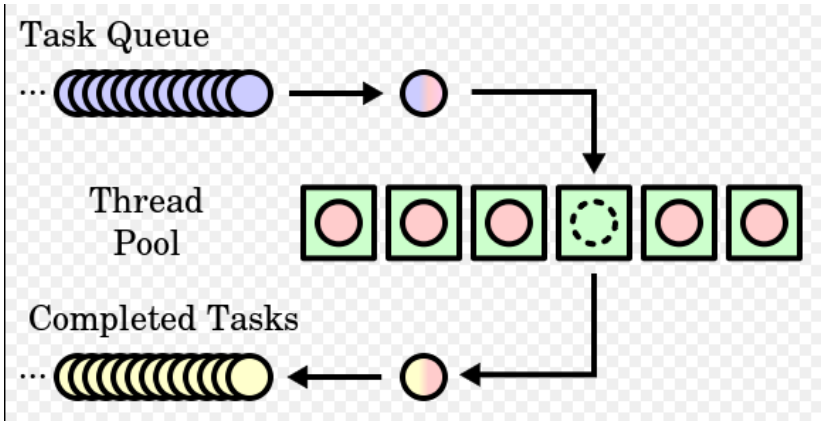


8.1.2. תבנית Thread Pool

תבנית Tread Pool זה כאשר מספר תהליכונים נוצרים כדי לבצע מספר משימות, אשר בדרך כלל מאורגנים בתור. התוצאות לאחר ביצוע המשימות יכולות להיות גם כן מאורגנות בתור, או המשימות יכולות להחזיר שום תוצאה. בדרך כלל יש יותר משימות מאשר תהליכונים. ברגע שתהליכון סיים את משימתו הוא יבקש משימה הבאה מהתור כל עוד יש משימות בתור. התהליכון יכול להיעצר או ללכת לישון עד שלא יהיו משימות חדשות בתור.
מספר של תהליכונים אשר בשימוש יכולים להינתן כפרמטר כדי לתת יעילות מרבית. בנוסף מספר של תהליכונים יכול להשתנות בצורה דינאמית לפי מספר משימות הממתינות.
מחיר של Thread pool גדול הוא שיש יותר שימוש במשאבי המחשב. האלגוריתם שבו משתמשים להגדרת מתי ליצור או להרוס תהליכון ישפיע על הביצועים הכללים:

- יצירה רבה של תהליכונים גורמת לבזבוז של משאבים לבזבוז זמן ליצירת תהליכונים לא נחוצים.
- השמדה רבה של תהליכונים גורמת לבזבוז זמן מאוחר יותר כדי ליצור שוב תהליכונים.
- יצירת תהליכונים איטית עלולה לגרום לביצועים נמוכים (זמן המתנה ארוך).
- השמדת תהליכונים בצורה איטית יכולה לגרום להרעבה של תהליכים אחרים.

בחירת האלגוריתם יהיה תלוי בסוג הבעיה, אם מספר המשימות מאוד גדול אז יצירת תהליכון עבור כל אחד מהם לא יעיל.
היתרון בשימוש של Thread Pool מול יצירת תהליכון חדש לכל משימה הוא שזמן יצירה והשמדה של תהליכון לא נכלל, מה שגורם לביצועים טובים יותר ומערכת יותר יציבה.



בפרויקט שלנו בבדיקת חברים על מנת להתחבר למחשב אנו צריכים לבצע התחברות מה שלעיתים יכול להיכשל בגלל מספר סיבות. עקב כך יצרנו 3 ניסיונות של חיבור. במידה וההתחברות נכשלה זה גורם להמתנה רבה, לכן כאשר יש מספר רב של מחשבים שאליהם צריך להתחבר, אם נעשה זאת בתור, זמן ההמתנה של הבדיקה יהיה ארוך מאוד. מסיבה זאת אנו משתמשים ב - Thread Pool על מנת ליעל את זמן הבדיקה ע"י כך שכל ניסיונות ההתחברות יהיו מקבילות.

9. בדיקות מערכת

9.1. מטרת הפרק

הגדרת תוכנית בדיקות למערכת ויישומה , על מנת לבדוק האם ישנם כשלים במערכת והאם התוכנה מבצעת את מה שהוגדרה לבצע במסמך הדרישות.
פרק זה יחולק לשני חלקים:

- תכנון הבדיקות– (Software Test Plan) **STP**
- ביצוע הבדיקות והצגת התוצאות – **STD + STR**
  (Software Test Details + Software Test Results)

9.2. תיאור המערכת

המערכת הינה תוכנה ייעודית לזיהוי התחברות לרשת בזמן אמת, אשר מבצעת פעולות לניטור הרשת וביצוע פעולות נלוות. תתי מערכות :

- מערכת זיהוי רשת.
- מערכת לביצוע פעולות נלוות .

9.3. שלב תכנון הבדיקות STP

מטרת חלק זה היא להוות מסגרת לביצוע הבדיקות , בהתחשב במגבלות כ"א, עלויות וזמן. זהו החלק בו נקבעות ונבנות הבדיקות אשר יענו בצורה הטובה ביותר על השאלה האם המערכת מתפקדת בהתאם לדרישות שהוגדרו והאם ראוי לקבל את גרסה זו של המערכת.

9.3.1. אסטרטגיית הבדיקות

- **<u>בדיקות פונקציונאליות:</u>**
  ביצוע בדיקות לדרישות הפונקציונאליות שהוגדרו במסמך הדרישות . בדיקת פעולות בדידות המתבצעות במערכת , על ידי המשתמש או באופן עצמאי של המערכת .
- **<u>בדיקת ממשק משתמש:</u>**
  בדיקת ממשק המשתמש בתוכנה . בדיקת תהליכים שנוצרים על ידי פעולה כלשהיא של המשתמש.
- **<u>בדיקת פעולות אוטומטיות של המערכת:</u>**
  בדיקת פעולות אוטומטיות שהתוכנה מבצעת ברקע , לא מופעלות על ידי בקשת המשתמש .

9.3.2. תיחום הבדיקות

- **<u>בדיקת GUI (ממשק משתמש גרפי)</u>** בדיקות אלו בודקות את התנהגות המערכת - באינטראקציה עם המשתמש .
- **<u>בדיקת גבולות ותחומים</u>** בדיקות אלו בודקות את התנהגות המערכת במתן ערכים בתוך התחום הרצוי ומחוץ לתחום הרצוי .
- **<u>בדיקות ביצועים</u>** בדיקות אלו בודקות האם המערכת מתפקדת כצפוי מבחינת זמן התגובה שלה .
- **<u>בדיקות פונקציונאליות</u>** נוספות יתבצעו על ידי פעולות קלט שונות (תגובות , פעולות)

9.3.3. דרישות לביצוע הבדיקות

הבדיקות יוכלו להתבצע אך ורק ברשת ארגונית, מפני שרק ברשת ארגונית התוכנה תכיל את כל הרכיבים הנחוצים לפעולה תקינה. כמו כן צריך לפחות 2 מחשבים לבדיקה מסוימת.

9.4. STD+STP

9.4.1. בדיקות ממשקי משתמש בדיקות – GUI

<u>בדיקת כפתורים ראשיים :</u>

זיהוי הבדיקה	01-01		
מטרת הבדיקה	בדיקת התקנת תצורה של Task Scheduler		
תנאים מקדימים	אין		
צעד לביצוע	**תיאור הצעד**	**תוצאה צפויה**	**סטאטוס**
1.	לחיצה כפולה על קובץ ההתקנה	הופעה של שורת פקודה חדשה ב - Task Scheduler	עבר

זיהוי הבדיקה	02-01		
מטרת הבדיקה	בדיקת תצורה של קובץ RecognizeAction.txt		
תנאים מקדימים	מפעילים את ה GUI בלשונית check		
צעד לביצוע	**תיאור הצעד**	**תוצאה צפויה**	**סטאטוס**
1.	מסמנים ב V ליד Gateways ממלאים את השדה 10.0.0.138 ולוחצים על Add	10.0.0.138 נמצא ברשימת הגלילה	עבר
2.	לוחצים על כפתור Save	תופיע הודעה נשמר	עבר
3.	פותחים את הקובץ ובודקים שיש שורה הבאה on;Gateway;10.0.0.138;	on;Gateway;10.0.0.138;	עבר

זיהוי הבדיקה	02-02		
מטרת הבדיקה	בדיקת תצורה של קובץ RecognizeAction.txt		
תנאים מקדימים	מפעילים את ה GUI בלשונית check		
צעד לביצוע	**תיאור הצעד**	**תוצאה צפויה**	**סטאטוס**
1.	מסמנים ב V ליד Ip Ranges ממלאים את השדה 10.0.0.1-255 ולוחצים על Add	10.0.0.1-255 נמצא ברשימת הגלילה	עבר
2.	לוחצים על כפתור Save	תופיע הודעה נשמר	עבר
3.	פותחים את הקובץ ובודקים שיש שורה הבאה on;IpRange;10.0.0.1-255;	on;IpRange;10.0.0.1-255;	עבר

זיהוי הבדיקה	02-03		
מטרת הבדיקה	בדיקת תצורה של קובץ RecognizeAction.txt		
תנאים מקדימים	מפעילים את ה GUI בלשונית check		
צעד לביצוע	**תיאור הצעד**	**תוצאה צפויה**	**סטאטוס**
1.	מסמנים ב V ליד Websites ממלאים את השדה web.ifd.net ולוחצים על Add	web.ifd.net נמצא ברשימת הגלילה	עבר
2.	לוחצים על כפתור Save	תופיע הודעה נשמר	עבר
3.	פותחים את הקובץ ובודקים שיש שורה הבאה on;Website;web.ifd.net;	on;Website;web.ifd.net;	עבר

זיהוי הבדיקה	02-04		
מטרת הבדיקה	בדיקת תצורה של קובץ RecognizeAction.txt		
תנאים מקדימים	מפעילים את ה GUI בלשונית check		
צעד לביצוע	**תיאור הצעד**	**תוצאה צפויה**	**סטאטוס**
1.	מסמנים ב V ליד Domain Name ממלאים את השדה idf.net ולוחצים על Add	idf.net נמצא ברשימת הגלילה	עבר
2.	לוחצים על כפתור Save	תופיע הודעה נשמר	עבר
3.	פותחים את הקובץ ובודקים שיש שורה הבאה on;DomainName;idf.net;	on;DomainName;idf.net;	עבר

זיהוי הבדיקה	02-05		
מטרת הבדיקה	בדיקת תצורה של קובץ RecognizeAction.txt		
תנאים מקדימים	מפעילים את ה GUI בלשונית check		
צעד לביצוע	**תיאור הצעד**	**תוצאה צפויה**	**סטאטוס**
1.	מסמנים ב V ליד Friends ממלאים את השדות Network code : 123 Number friend : 2 Delay between checks : 1	אין	עבר
2.	לוחצים על כפתור Save	תופיע הודעה נשמר	עבר
3.	פותחים את הקובץ ובודקים שיש שורה הבאה on;Friends;2;1;	on;Friends;2;1;	עבר

זיהוי הבדיקה	02-06		
מטרת הבדיקה	בדיקת תצורה של קובץ MakeAction.txt		
תנאים מקדימים	מפעילים את ה GUI בלשונית Action		
צעד לביצוע	**תיאור הצעד**	**תוצאה צפויה**	**סטאטוס**
1.	מסמנים ב V ליד Email to report ממלאים את השדה vladi@gmail.com ולוחצים על Add	vladi@gmail.com נמצא ברשימת הגלילה	עבר
2.	לוחצים על כפתור Save	תופיע הודעה נשמר	עבר
3.	פותחים את הקובץ ובודקים שיש שורה הבאה on;SendEmail; vladi@gmail.com;	on;SendEmail; vladi@gmail.com;	עבר

זיהוי הבדיקה	02-07		
מטרת הבדיקה	בדיקת תצורה של קובץ MakeAction.txt		
תנאים מקדימים	מפעילים את ה GUI בלשונית Action		
צעד לביצוע	**תיאור הצעד**	**תוצאה צפויה**	**סטאטוס**
1.	מסמנים ב V ליד Disable Network	אין	עבר
2.	לוחצים על כפתור Save	תופיע הודעה נשמר	עבר
3.	פותחים את הקובץ ובודקים שיש שורה הבאה on;DisableNetwork;	on;DisableNetwork;	עבר

זיהוי הבדיקה	02-08		
מטרת הבדיקה	בדיקת תצורה של קובץ MakeAction.txt		
תנאים מקדימים	מפעילים את ה GUI בלשונית Action		
צעד לביצוע	**תיאור הצעד**	**תוצאה צפויה**	**סטאטוס**
1.	מסמנים ב V ליד Shutdown	אין	עבר
2.	לוחצים על כפתור Save	תופיע הודעה נשמר	עבר
3.	פותחים את הקובץ ובודקים שיש שורה הבאה on;Shutdown;	on;Shutdown;	עבר

זיהוי הבדיקה	02-09		
מטרת הבדיקה	בדיקת תצורה של קובץ MakeAction.txt		
תנאים מקדימים	מפעילים את ה GUI בלשונית Action		
צעד לביצוע	**תיאור הצעד**	**תוצאה צפויה**	**סטאטוס**
.1	מסמנים ב V ליד Logout	אין	עבר
.2	לוחצים על כפתור Save	תופיע הודעה נשמר	עבר
.3	פותחים את הקובץ ובודקים שיש שורה הבאה ;on;Logout	;on;Logout	עבר

זיהוי הבדיקה	02-10		
מטרת הבדיקה	בדיקת תצורה של קובץ MakeAction.txt		
תנאים מקדימים	מפעילים את ה GUI בלשונית Action		
צעד לביצוע	**תיאור הצעד**	**תוצאה צפויה**	**סטאטוס**
.1	מסמנים ב V ליד Lock	אין	עבר
.2	לוחצים על כפתור Save	תופיע הודעה נשמר	עבר
.3	פותחים את הקובץ ובודקים שיש שורה הבאה ;on;Lock	;on;Lock	עבר

זיהוי הבדיקה	02-11		
מטרת הבדיקה	בדיקת תצורה של קובץ Policy.txt		
תנאים מקדימים	מפעילים את ה GUI בלשונית Policy		
צעד לביצוע	**תיאור הצעד**	**תוצאה צפויה**	**סטאטוס**
.1	בוחרים בפעולה Lock מעמודה השמאלית ומסמנים בדיקות 1 ו 3 מעמודה ימנית ולוחצים על Add	נוספת שורה חדשה ברשימת גלילה Lock.1.3	עבר
.2	לוחצים על כפתור Save	תופיע הודעה נשמר	נכשל
.3	פותחים את הקובץ ובודקים שיש שורה הבאה Lock;1;3	Lock;1;3	נכשל

זיהוי הבדיקה	02-12		
מטרת הבדיקה	בדיקת תצורה של קובץ Policy.txt		
תנאים מקדימים	מפעילים את ה GUI בלשונית Policy ממלאים ושומרים את הלשוניות Check ו Action לפני שעוברים ללשונית Policy		
צעד לביצוע	**תיאור הצעד**	**תוצאה צפויה**	**סטאטוס**
1.	בוחרים בפעולה Lock מעמודה השמאלית ומסמנים בדיקות 1 ו 3 מעמודה ימנית ולוחצים על Add	נוספת שורה חדשה ברשימת גלילה Lock.1.3	עבר
2.	לוחצים על כפתור Save	תופיע הודעה נשמר	עבר
3.	פותחים את הקובץ ובודקים שיש שורה הבאה Lock;1;3	Lock;1;3	עבר

בדיקת פונקציונאליות של התוכנה :

זיהוי הבדיקה	03-01		
מטרת הבדיקה	פעילות תקינה של בדיקת Gateway והפעלת פעולה Send Email		
תנאים מקדימים	הוספה ל Check בדיקה בשם Gateway , הוספה ל Action פעולה בשם Send Email , הוספה ל Policy חוק המקשר בין הפעולה לבדיקה		
צעד לביצוע	**תיאור הצעד**	**תוצאה צפויה**	**סטאטוס**
1.	מפעילים את כרטיס הרשת ומתחברים לרשת	נשלח Email	עבר

זיהוי הבדיקה	03-02		
מטרת הבדיקה	פעילות תקינה של בדיקת Ip Range והפעלת פעולה Disable Network		
תנאים מקדימים	הוספה ל Check בדיקה בשם Ip Range, הוספה ל Action פעולה בשם Disable Network , הוספה ל Policy חוק המקשר בין הפעולה לבדיקה		
צעד לביצוע	**תיאור הצעד**	**תוצאה צפויה**	**סטאטוס**
1.	מפעילים את כרטיס הרשת ומתחברים לרשת	ניתוק מהרשת	עבר

זיהוי הבדיקה	03-03		
מטרת הבדיקה	פעילות תקינה של בדיקת Website והפעלת פעולה Shutdown		
תנאים מקדימים	הוספה ל Check בדיקה בשם Website, הוספה ל Action פעולה בשם Shutdown , הוספה ל Policy חוק המקשר בין הפעולה לבדיקה		
צעד לביצוע	**תיאור הצעד**	**תוצאה צפויה**	**סטאטוס**
1.	מפעילים את כרטיס הרשת ומתחברים לרשת	כיבוי מחשב	עבר

זיהוי הבדיקה	03-04		
מטרת הבדיקה	פעילות תקינה של בדיקת Domain Name והפעלת פעולה Logout		
תנאים מקדימים	הוספה ל Check בדיקה בשם Domain Name, הוספה ל Action פעולה בשם Logout, הוספה ל Policy חוק המקשר בין הפעולה לבדיקה		
צעד לביצוע	**תיאור הצעד**	**תוצאה צפויה**	**סטאטוס**
1.	מפעילים את כרטיס הרשת ומתחברים לרשת	יציאה מהמשתמש	עבר

זיהוי הבדיקה	03-05		
מטרת הבדיקה	פעילות תקינה של בדיקת Friends והפעלת פעולה Lock		
תנאים מקדימים	הוספה ל Check בדיקה בשם Friends, הוספה ל Action פעולה בשם Lock , הוספה ל Policy חוק המקשר בין הפעולה לבדיקה		
צעד לביצוע	**תיאור הצעד**	**תוצאה צפויה**	**סטאטוס**
1.	מפעילים את כרטיס הרשת ומתחברים לרשת	נעילת מחשב	עבר

10. המלצות לפיתוח עתידי

פרויקט זה היה מאוד מאתגר עבורנו והוא מהווה אב טיפוס ,ואנו חושבים שמי שירצה להמשיך לפתח את הפרויקט ימצא אותו למאוד מעניין ושימושי.
חשבנו על כמה כיווני פיתוח אפשריים לפרויקט שלנו:

- ניתן לשפר את התוכנה להיות Daemon על מנת לעבוד מבלי להיחשף למשתמש במחשב בו מותקנת התוכנה.
- להרחיב את הפעולות שהתוכנה יכולה לעשות לדוגמא:
  - שמירת תעבורת רשת TCP/UDP.
  - לעקוב אחר פעולות המשתמש .
  - הצפנת חומר חשוב הנמצא באותו מחשב.
- להרחיב את הבדיקות שהתוכנה יכולה לעשות בהתאם לארגון.

11. סקר ספרות

11.1. הקדמה

<u>מה ההבדל בין רשת Client/Server לבין רשת Peer-to-Peer?</u>

יש הבדל עצום בין Client/Server ורשתות Peer-to-Peer. לדוגמא, לרשת Peer-to-Peer אין שרת מרכזי. כל תחנת עבודה ברשת משתפת את הקבצים שלה באופן שווה עם אחרים. אין אחסון או אימות של משתמשים מרכזיים. לעומת זאת, לרשת Client/Server יש שרתים נפרדים ייעודיים ולקוחות ברשת. דרך תחנות עבודה של הלקוח, משתמשים יכולים לגשת לקבצים אשר בדרך כלל מאוחסנים בשרת. השרת יקבע אילו משתמשים יכולים לגשת לקבצים ברשת.

- <u>רשתות Peer-to-Peer</u>

רשתות Peer-to-Peer מתאימות רק לעסקים קטנים מאוד או לשימוש ביתי. רשת Peer-to-Peer יכולה לתמוך כעשר לקוחות (תחנות עבודה) לפני שהיא מתחילה לסבול מכמה בעיות ביצועים וניהול רציניות.
הרעיון מאחורי רשת Peer-to-Peer הוא לשתף קבצים ומדפסות באופן הזול ככל האפשר. לכן, אין שרת מרכזי ברשת. במקום זאת, כל פונקציות לקוח הן כלקוח וכשרת בו-זמנית. כל המשתמשים מורשים לשלוט בגישה למשאבים במחשבים שלהם, לכן ביטחון מידע הופך להיות מאוד מסוכן ברשת זו. הביטחון ברשת Peer-to-Peer הוא רק ברמת השיתוף עצמו . כאשר משתמשים יוצרים שיתוף ברשת, הם יכולים לתת גישה לכל אחד, מה שאומר שלכל אחד תהיה גישה מלאה למשאב. כמו כן הם יכולים להגדיר סיסמא לשיתוף, מה שאומר שרק מי שמכיר סיסמא זו יוכל לגשת למשאבים אלו.

- <u>רשתות Client/Server</u>

יש מגוון רב של רשתות Client/Server, אבל לכולם יש כמה דברים במשותף. דבר אחד בטוח, לכולם יש מסדי נתונים של אבטחה השולטים בגישה למשאבים משותפים בשרתים. השרת מכיל רשימה של שמות משתמשים וסיסמאות. משתמשים אינם יכולים להיכנס לרשת, רק בתנאי שהם מספקים שמות משתמש וסיסמאות בתוקף לשרת. ברגע שמשתמש נכנס למערכת, הוא יכול לגשת רק למשאבים שמנהל הרשת מאפשר להם גישה. כך, רשתות Client/Server נותנות הרבה יותר ביטחון מאשר רשתות Peer-to-Peer. החיסרון העיקרי לרשת Client/Server הוא העלות שלה. שרתים יכולים להיות מאוד יקרים, וכן לאנשי תמיכה ברשת צריכה להיות הכשרה איך לנהל רשת זו.

- <u>באיזה רשת אנו נתעסק בפרויקט ?</u>

בפרויקט שלנו נתעסק עם רשת מסוג Client/Server.
נסביר איזה בעיה קיימת ברשת זו ונדון על הפתרון שלה בעזרת תכונות של הרשת.

11.2. תיאור הבעיה

האבטחה ברשת Client/Server נועדה לשמור על המשאבים של השרת ,איך אנו נגן על החומר ששמור במחשב עצמו?

<u>דוגמאות לאירועים המתארים את הבעיה</u> :

- משתמש נכנס לרשת עם שם משתמש וסיסמא תקינים, יש לו גישה לקבצים של הרשת. הוא ניגש לקבצים מהשרת, מוריד אותם על מחשב. לאחר זאת בעזרת כבל רשת מחבר מחשב זה למחשב נייד, ולמרות שהוא כבר לא מקושר לרשת שלנו, הוא עדיין נמצא במחשב ויכול להעביר קבצים אשר הוריד מהשרת למחשב שאיננו מחשב צבאי.
- משתמש נכנס לרשת עם שם משתמש וסיסמא תקינים במחשב נייד, עובד על קבצים שלו במחשב , אשר בעתיד ישים אותם על השרת. המשתמש הלך למשרד של חבר שלו, ומבלי לשים לב, המחשב התנתק מהרשת הצבאית ותחבר לרשת שונה. במצב זה, הקבצים שעל המחשב עלולים להיות בסכנה , כיוון שהוא מחובר לרשת פתוחה.
- מנהלי רשת העבירו בטעות מחשב למקום אחר בצבא מבלי לפרמט אותו. המחשב מקבל הגדרות רשת בצורה דינמית, לכן לאחר ההעברה יקבל הגדרות שרת המתאימות לרשת לאן שהעבירו מחשב זה. משתמשים יכולים להיכנס למחשב זה בעזרת שמות משתמש וסיסמא שלהם כרגיל ולראות את המידע שעל המחשב שלא אמורים לראות.

לפי דוגמאות שנתנו ניתן לראות שלא מספיק להגן רק על המשאבים שנמצאים על השרת ויש צורך בהגנה נוספת על המחשב עצמו, כדי שמידע שכבר נמצא על המחשב לא ידלוף החוצה. לשם פתרון בעיה זו אנו נתמקד בעיקר בשני דברים :

- זיהוי חיבור לרשת זרה .
- איך להגן על המשאבים של המחשב או לאסוף מידע על פעולות של המשתמש.

11.3. השיטה

בהתקנת התוכנה יוגדרו הגדרות שתוכנה תבדוק איתם מאפיינים של רשת ואיזה פעולות לבצע במידה ויש התחברות לרשת זרה.

11.4. בדיקות היכולות להעיד על חיבור לרשת זרה

יתבצעו בו זמנית מספר בדיקות אשר יכולות להעיד על כך שאנו ברשת זרה. למערכת יהיו מספר בדיקות לשם אבטחה במידה והמשתמש מכיר איזה הגדרות אמורות להיות לרשת התקינה.

הבדיקות הן :

1) <u>בדיקת IP של המחשב</u> – זוהי בדיקה שמתבצעת ע"י CMD . הבדיקה תבדוק האם ה IP של המחשב מוגדר כתקין למחשב זה לפי הגדרות שהוגדרו לתוכנה.
2) <u>בדיקת Default Gateway</u> – זוהי בדיקה שמתבצעת ע"י CMD . הבדיקה תבדוק האם שער ברירת של המחשב הינו תקין ושאין חיבור נוסף למחשב (לדוגמה בכבל רשת) לרשת נוספת.
3) <u>בדיקת Ping</u> – הבדיקה תבדוק האם יש קשר לאתר כלשהו שאמור להיות רק ברשת זו. במידה והאתר איננו זמין זה יכול להעיד על 2 דברים : אנו ברשת זרה או שהאתר לא זמין. לכן הבדיקה היא מספקת אם יש גישה, אך במידה ואין גישה צריך לבצע בדיקות נוספות.

4) בדיקת חברים לרשת – הבדיקה שולחת מסרון מוצפן למספר משתמשים שמחוברים כרגע לרשת זו (במידה והם מחוברים) אשר מכיל קוד סודי של הרשת התקינה. המחשבים שמקבלים הודעה זו יוכלו לפתוח אותה רק במידה ומותקנת אצלם התוכנה ורק התוכנה תוכל לפענח את המסרון כדי לבדוק האם הקוד הסודי תואם לקוד הסודי שהם מכירים. למצב זה יש מספר אפשרויות שיקרו :
   - במידה והקוד הסודי תואם - זה אומר שהמחשב נמצא ברשת תקינה, לכן יחזירו תשובה למחשב שבודק את עצמו שהכל תקין.
   - במידה והקוד הסודי לא תואם – זה אומר שהמחשב נמצא ברשת לא תקינה, לכן יחזירו תשובה למחשב שבודק את עצמו שהוא נמצא לא ברשת התקינה.
   - אין תוכנה על המחשב שאליו שולחים את המסר – כיוון שאין תוכנה שתפענח את המסרון לכן לא תוחזר אף תשובה. לכן המחשב אשר בודק את עצמו יציב זמן מוגדר לקבלת תשובה, במידה ולא קיבל תשובה בזמן זה, הוא יבין שהוא נמצא ברשת זרה.
5) בדיקת DNS - הבדיקה תשלח חבילה לשרת ותבדוק האם השרת חיי, במידה ויש חיבור לרשת אחרת , שרת זה לא יהיה זמין. כמו כן בעת התחרות מחשב לרשת נוספת, המחשב מקבל DNS מתאים, נבדוק האם ה - DNS שקיבל אכן מורשה לפי הגדרות התוכנה.
6) בדיקת Domain - כאשר משתמש מתחבר לרשת עם שם משתמש וסיסמא, המחשב מקבל הגדרות לפי Domain שאליו התחבר. נבדוק Domain שהתחבר למחשב אכן מורשה בהגדרות של התוכנה. במידה והמחשב יתחבר למקום נוסף, אנו נבדוק שה Domain נגיש בכל רשת אשר המחשב מחובר אליה. במידה והשרת Domain אינו זמין באחת מהרשתות זה מעיד על חיבור לרשת זרה.

11.5. זיהוי התחברות לרשת

פעולה זו מזהה שינוי כלשהו במחשב לגביי חיבורי רשתות, כלומר האם קרתה התחברות לרשת נוספת כלשהי. לדוגמה חיבור כבל רשת, התחברות אלחוטית , חיבור דרך Net Stick וכדומה.
קיימות מספר טכניקות למימוש זיהוי זה:

1. בדיקה של לוגים אשר מוטמנים במערכת ההפעלה, הרי שכל שינוי קטן ופעולה נרשמים בלוגים של מערכת ההפעלה ולכן ניתן לאתר את השינוי ומתי בדיוק הוא נוצר.
   - חסרון - גילינו שיש יותר מידי סוגים של לוגים, והם גם פחות מובנים , וכמו כן איתור בעזרתם נורא מסורבל. בתוספת בשיטה זו צריך להוסיף תוכנת עזר ולהיעזר בה כדי לגשת ללוגים. כמו כן ייווצר עומס בתוכנה כיוון שיש צורך ליצור לולאה אשר תרוץ כל הזמן על הלוג ותבדוק האם התווספה שורה חדשה המעידה על שינוי ברשת. מעבר לבעיות אשר הודגמו יש צורך גם כן לחקור על אופן כתיבתם של הלוגים, על מנת לדעת איך לעבוד אם המידע בתוך הלוג.

2. <u>מעקב אחר שינויים ברשת ע"י DEAMON</u>. זהו תהליך אשר רץ כל הזמן ברקע ובודק האם יש שינויים. לדוגמה כתובת IP השתנתה, או רכיב רשת מסוים קיבל הגדרות רשת שזה מעיד על חיבור נוסף דרך הרכיב הזה.
   - <u>חסרון</u> – בשיטה זו יש צורך ליצור תוכנת בדיקה אשר רצה כל הזמן ברקע ובודקת כל פרק זמן קצר האם יש התחברות לרשת נוספת. בדיקות אלו דורשות משאבים מהמעבד ולכן בדיקות אלו עלולות ליצור עומס על המערכת ההפעלה.
3. <u>בדיקה ע"י Task Scheduler</u>. זוהי תוכנה של מערכת Windows. בעזרת Task Scheduler ניתן לייצר משימה שתתפוס חיבור כלשהו לרשת ותפעיל את התוכנה שלנו לשם ביצוע בדיקות נדרשות.

<u>הטכניקה הנבחרת:</u>

בחרנו בשיטה 3 -

בחרנו בשיטה זו כיוון היא עונה על החסרונות של שיטות אחרות.

- התוכנה הנ"ל היא מובנת במערכת ההפעלה של Windows . היא מבצעת האזנה תמידית לאירועים ולא גורמת עומס יתר על המחשב . כמו כן בעזרת שיטה זו לא נעמיס על הרשת, כיוון שנבצע בדיקות אך ורק כאשר יש התחברות לרשת נוספת . לכן היא עונה על חסרון של שיטה מספר 2.

תוכנה זו לא מחפשת אירוע שקרה בלוג, אלה התוכנה נרשמת לאירוע מסוים. ברגע שהאירוע מתרחש התוכנה מבצעת את המשימה שהוטלה עליה ורק לאחר מכן נרשם האירוע בתוך הלוג. מסיבה זו אין צורך לחפש התחברות לרשת דרך הלוגים, אלה עדיף להירשם לאירוע "התחרות לרשת". לכן השיטה עונה גם כן על החיסרון של שיטה מספר 1.

11.6. הגנת המחשב במידה והמחשב ברשת זרה

למשימה זו יש מספר אופציות אשר ניתן לבצע וזאת מגדירים בהתקנת התוכנה. האופציות הן:

- כיבוי מחשב – זוהי פעולה המתבצעת בעזרת פקודת CMD שמכבה את המחשב . במידה ויפעילו מחשב זה והוא עוד פעם יתחבר לרשת זרה , המחשב כל פעם יכבה את עצמו.
- נעילת מחשב – זוהי פעולה אשר לא מפריעה לפעולות שנעשו על המחשב לפני שתחבר לרשת זרה, כיוון שאם משתמש מתחבר חזרה כל הקבצים הפתוחים והתוכנות יחזרו להיות כמו קודם. אך איננה נותנת למשתמש להמשיך לעבוד הלאה על המחשב כל עוד הוא מחובר לרשת זרה.
- יציאה מהמשתמש – זוהי פעולה אשר מנתקת את המשתמש מהמחשב , וכל עוד הוא ינסה להיכנס חזרה למחשב אשר עדיין מחובר לרשת זרה, הוא יתנתק שוב פעם.
- שליחת\שמירת דוח – בעת התחברות לרשת הזרה התוכנה תאסוף פרטים על הרשת הזרה ועל המשתמש אשר עשה זאת על מנת להתריע לגורמים רלוונטיים .
- חסימת רשת – על מנת לא לפגוע במשך העבודה של המשתמש, ועל מנת שיוכל להמשיך לעבוד כרגיל, התוכנה תשאיר את המשתמש בתוך המחשב אך תחסום גישה לרשת הזרה ע"י נעילת כרטיס הרשת, כדי שהמידע לא ידלוף לרשת הזרה.
- הקלטת תעבורה – על מנת לעקוב אחר פעולות של המשתמש , נאסוף מידע לגביי תעבורת רשת, זיכרון המחשב ואילו תוכנות היו פועלות ברקע במערכת ההפעלה.

12. ביבליוגרפיה

[1] ספר לתכנות ברשתות

Fundamental Networking in Java , Esmond Pitt, 2006.

[2] ספר לשיתוף ברשת

Peer-to-peer Systems, Ralf Steinmetz, Klaus Wehrle, 2005.

[3] טכניקות לסינון כתובות ברשת

https://wiki.xtronics.com/index.php/IP_Subnet_Masks , Transtronics, Inc, 2007.

[4] Task Scheduler עבודה עם

https://technet.microsoft.com/en-us/library/bb490996.aspx , Microsoft 2015.

http://www.groovypost.com/howto/automatically-run-script-on-internet-connect-network-connection-drop/ , Jack Busch ,2012.

[5] מבוא Java

http://javabook.co.il/wp/המהדורה-החדשה/ , Life Michael , 2009.