

	
בית ספר "אמירים" כפר ורדים	מגשימים – מוקד כרמיאל

## מסמך מסכם



**שם התלמיד:** שביט בוריסוב

**ת.ז.:** 318740354

**אחראי פרויקט:** יואב פוירשטיין

**מנטור:** מתי פיקוס

**תאריך:** 30.04.2016

היסטוריית גרסאות המסמך:	
תיאור	תאריך
התחלת העבודה	28.04.2016
סיום הקובץ	30.04.2016
מספר תוספות	31.05.2016

## תוכן עניינים

1. רקע .....	2
1.1. רציונל הפרויקט .....	2
1.2. מבוא .....	2
1.3. תיחום הפרויקט .....	2
2. תיאור שיווקי של המוצר המוגמר .....	3
3. סביבת העבודה .....	4
3.1. שפות תכנות .....	4
3.2. סביבת עבודה לפיתוח .....	4
3.3. סביבת עבודה לבדיקות .....	4
4. אלגוריתמים מרכזיים בפרויקט .....	5
4.1. ניסוח וניתוח הבעיה האלגוריתמית .....	5
4.2. אלגוריתמים קיימים לפתרון הבעיה .....	5
4.3. סקירת הפתרון הנבחר .....	5
5. ממשק המשתמש .....	6
5.1. התקנת המערכת .....	6
5.2. תפעול המערכת .....	6
6. רפלקציה אישית .....	9
7. המוצר המוגמר אל מול התכנון הראשוני .....	10
8. ביבליוגרפיה .....	11

## 1. רקע

### 1.1. רציונל הפרויקט

החלטתי לפתח את מערכת **MagShare** בכדי להעמיק את הידע שלי בתקשורת. מאז ומתמיד התעניינתי בתקשורת וספציפית באופן הפעולה של מערכות **Torrent**, ועל כן החלטתי לממש מערכת **Peer2Peer** בעצמי.

המערכת עונה על בעיית עומס השרתים אשר מתבצע בתקשורת **Client-Server** רגילה. במקום להעמיס על שרת אחד בקשות רבות מצד קליינטים, במערכת **Peer2Peer** זו מתבצעת תקשורת מפוזרת, שבמהלכה **Peers** שונים משתפים קובץ אחד בכל פעם ונותנים ל-**Peers** אחרים להוריד מהם קובץ זה. כך נמנע עומס על כל אחד מה-**Peers** וה-**Tracker**, אשר מקשר בין **Peers** שונים כדי שיתקשרו בינם לבין עצמם, שומר מעט מאוד נתונים אשר מאפשרים עבודה יעילה ומהירה.

כמו כן, מערכת **MagShare** מאפשרת שיתוף בטוח של קבצים, שכן היא בנויה על פרוטוקול ייחודי ולא מוכר אותו יצרתי. פרוטוקול זה הינו מוצפן והאפשרות ל"גניבת" נתונים הינה בעייתית ביותר.

באופן כללי, המערכת מאפשרת שיתוף בטוח ומהיר של קבצים, בין שני צדדים אשר בהכרח רוצים בכך.

### 1.2. מבוא

מערכת **MagShare** מבוססת על **Torrent**. במהלך הפיתוח קראתי רבות על הדרך בה **Torrent** עובד, ויצרתי מערכת הפועלת באופן דומה, אם כי מעט פשוטה יותר.

המערכת מורכבת משני רכיבים – **Client** ו-**Tracker**.

ה-**Client** אחראי על שיתוף (**Seeding**) והורדה (**Leeching**) של קבצים מ-**Peers** אחרים, אשר יודעים לעשות זאת גם כן. ממשיק המשתמש הינו דרך הקליינט, ובאמצעותו ניתן ליצור קבצי אפיון ייחודיים בשם קבצי **MSF** אשר מכילים מידע על הקבצים המיועדים להורדה, לערוך הגדרות וכו'.

ה-**Tracker** הינו רכיב שרץ על שרת כלשהו ותפקידו לקשר בין **Peers**. ה-**Tracker** מקבל בקשות להוריד קבצים מסוימים, מוצא באמצעות קבצי ה-**MSF** **Peers** אשר משתפים קבצים אלה ומחזיר רשימה שלהם, בכדי שתוכל להתבצע תקשורת **Peer2Peer**.

המערכת נכתבה בשפת **C++**, בסביבת העבודה **Qt**. היא משתמשת בטכנולוגיות מובנות של מערכות הפעלה הנתמכות על ידי **Qt**, כגון **socket**, מערכת קבצים, מנגנוני **UI** וכו'.

### 1.3. תיחום הפרויקט

מטרתה של המערכת לפתור את בעיית שיתוף הקבצים בדרכים שאינן מאובטחות, עקב תקשורות רגילות המאפשרות "גניבת" נתונים. באמצעות **MagShare** ניתן לשותף קבצים באופן מאובטח תוך שימוש בטכנולוגיית **Peer2Peer**, שאמצעותה ניתן לדעת בוודאות שמתקיימת תקשורת לגיטימית משני הצדדים.

הייחודיות של המערכת הינה בשימוש בפרוטוקול ייחודי (בשם **MSP**) אשר איננו מוכר לציבור ועל כן הינו בטוח מפני התקפות.

המערכת אינה מאפשרת הורדה מקבילה של מספר קבצים, שיתוף מקביל של מספר קבצים, שיתוף והורדה בו זמנית ועובדת אל מול **Tracker** אחד בלבד. על כן המערכת איננה מהווה פתרון לשיתוף והורדה של מספר קבצים בו בעת.

## 2. תיאור שיווקי של המוצר המוגמר

שם המוצר : MagShare.

MagShare הינה תכנת שיתוף קבצים הפועלת באמצעות Peer2Peer. באמצעות התכנה ניתן לשתף קבצים מכל הסוגים בין מחשבים שונים עליהם מותקנת התכנה, תוך היעזרות ב-Tracker. קהל היעד של התכנה הוא עבור כל משתמש ממוצע במחשב אשר ברצונו לשתף קבצים עם משתמשים אחרים או לחלופין להוריד קבצים אשר שותפו בידי משתמשים אחרים.

המערכת עונה על הצורך לשיתוף קבצים באופן מהיר ובטוח, בין שני צדדים הרוצים בכך. בניגוד למערכת Client-Server רגילה, במערכת MagShare כלל הצדדים המשתתפים בתקשורת הינם בוודאות אמינים, והקובץ המשותף/מתקבל הוא בהכרח הקובץ הרצוי. המערכת הינה פשוטה, נוחה לשימוש ומציעה פתרון אידיאלי לבעיה ספציפית.

בהשוואה לקליינטים שונים של Torrent (כגון uTorrent, BitTorrent וכו') מערכת MagShare פשוטה יותר – מתבצע בה שיתוף או הורדה של קובץ אחד בלבד בכל פעם, וניתן להתחבר ל-Tracker אחד בכל פעם. מצד אחד זהו חיסרון, שכן במערכות אחרות ניתן להוריד כמות גדולה של קבצים תוך שימוש בכמות רבה של Trackers ; עם זאת, היתרון בעיצוב זה הוא הפשטות שבשימוש במערכת, אשר מכוונת גם לאוכלוסיות שאינן בעלות ידע רב במחשבים (כגון ילדים, קשישים וכו').

להרחבת המוצר המוגמר ניתן להוסיף תמיכה ביותר מ-Tracker אחד, לאפשר הורדה של מספר קבצים במקביל ולאפשר שיתוף והורדה (של קבצים שונים) בו זמנית. שיפור אפשרי נוסף הוא בחירה חכמה יותר של Peers להורדה, על פי טיב התקשורת.

### 3. סביבת העבודה

#### 3.1. שפות תכנות

לצורך פרויקט זה השתמשתי בשפת C++. שפה זו הינה שפה ותיקה בעלת יכולות מרובות וכלים אינספור המציעה פתרונות לכלל הבעיות התכנותיות.

C++ הינה שפת Multi-Purpose ובחרתי בה משום שידעתי שאצטרך לשלב אלמנטים מתחומים שונים בתכנות בפרויקט שלי, והעדפתי לעשות זאת מבלי להשתמש ביותר משפת תכנות אחת.

סיבה נוספת בגללה בחרתי בשפת C++ היא העובדה ששפה זו הינה שפת Multi-Platform, וברצוני היה לפתח מערכת שתומכת ביותר מפלטפורמה אחת.

#### 3.2. סביבת עבודה לפיתוח

סביבת הפיתוח בה השתמשתי היא סביבת Qt. סביבה זו הינה סביבה ידועה הכוללת בתוכה Framework אשר מקל על כתיבת קוד מורכב. כמו כן, סביבה זו מאפשרת לקמפל ב-Multi-Platform ובשל רצוני לאפשר שימוש במערכת אותה יצרתי במספר פלטפורמות, עובדה זו היוותה יתרון.

בנוסף, ל-Qt קיימת קהילה רבת משתמשים ברחבי העולם, מה שאפשר שאלת שאלות ומציאת פתרונות לבעיות רבות בהן נתקלתי במהלך הדרך.

#### 3.3. סביבת עבודה לבדיקות

לא השתמשתי בכלים מיוחדים עבור בדיקות; השתמשתי בבדיקות ידניות, בין היתר באמצעות מערכת logging נרחבת אשר שילבתי במערכת.

## 4. אלגוריתמים מרכזיים בפרויקט

### 4.1. ניסוח וניתוח הבעיה האלגוריתמית

בכדי לאפשר העברת נתונים בין Peers, יש לחלק את הקובץ המיועד לשיתוף למספר חלקים וליצור Hash מוצפן של חלקים אלו (כדי לאפשר ל-Tracker למצוא את הקבצים המיועדים להורדה, פעולה אשר הוא מבצע באמצעות ה-Hash).

כדי לבצע תהליך זה, כמו גם בכדי לאפשר ל-Tracker לשמור מאגר של Peers פעילים, יש להשתמש באלגוריתם חכם המאפשר זאת.

### 4.2. אלגוריתמים קיימים לפתרון הבעיה

אלגוריתם מקובל לחילוק קובץ למספר חלקים מבצע שימוש בעץ בינארי בכדי לעשות זאת. ב-Qt, העץ הבינארי מיוצג באובייקט QMap. שמירת מאגר הנתונים של ה-Tracker בזיכרון מתבצעת גם היא תוך שימוש בעץ בינארי ובאובייקט QMap.

ליצירת Hash, ניתן להשתמש באלגוריתם הקריפטוגרפי SHA1. האלגוריתם מיושם ב-Qt על ידי האובייקט QCryptographicHash.

פתרונות אלו נלקחו מהדוקומנטציה של Qt ומהמסמך של BitTorrentProtocol.

### 4.3. סקירת הפתרון הנבחר

הפתרון בו השתמשתי בפועל הוא הפתרון המוצג לעיל. זהו הפתרון היחיד שמצאתי לבעיה, אשר היה בנוסף מתועד במסמך BitTorrentProtocol בו השתמשתי רבות במהלך הפרויקט ועל כן ידעתי שהינו פתרון אמין ונכון.

## 5. ממשק המשתמש

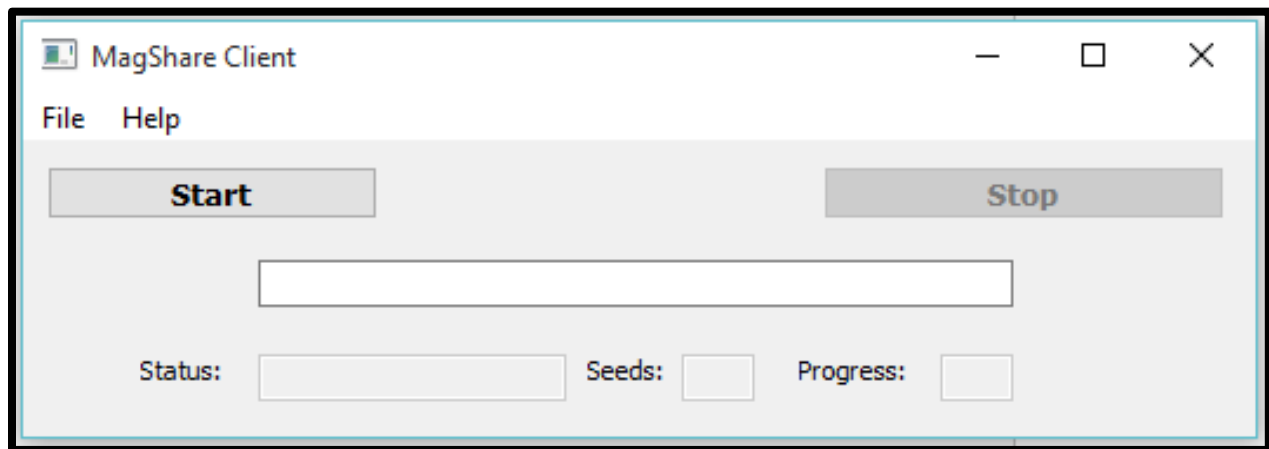
### 5.1. התקנת המערכת

יצירת ערכת התקנה של המערכת הייתה מחוץ לתחום הפרויקט, אולם עדיין ניתן להתקין את המערכת על כל מחשב העונה על הדרישות.

בכדי להתקין את המערכת, יש להוריד Qt Runtime למערכת ההפעלה המיועדת להריץ את התכנה (Linux או Windows). יש לוודא שהתקיימה בה נשמר Qt Runtime נמצאת ב-Path של מערכת ההפעלה. לאחר מכן יש לוודא שהתקיימה בה נמצאים קבצי ההרצה (וקבצי ההרצה עצמם) בעלי הרשאות מלאות. מומלץ למקם את קבצי ההרצה בתקיימה של Qt Runtime.

### 5.2. תפעול המערכת

בכדי לתפעל את המערכת, יש ראשית כל להפעיל את הקליינט. המסך הראשי של הקליינט נראה כך :

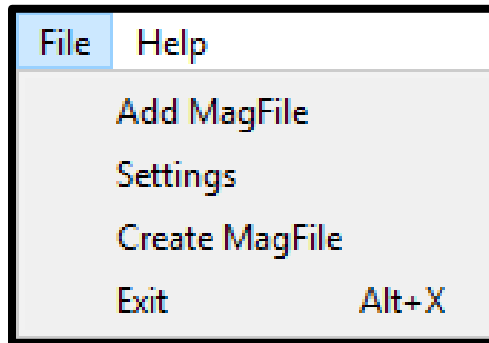


במסך זה ניתן להתחיל Leeching או Seeding של קובץ, על ידי לחיצה על כפתור Start לאחר בחירה של קובץ MSF. ניתן גם לעצור תהליך זה על ידי לחיצה על כפתור Stop. אם המערכת מזהה שהקובץ אליו מפנה ה-MSF כבר נמצא במערכת הקבצים (בתקיה שהוגדרה בדף הגדרות, מפורט בהמשך), מתבצע Seeding ; אחרת – מתבצע Leeching.

לאחר תחילת התהליך, מתעדכן שדה ה-Status מול ה-Tracker (אפשרויות שונות ל-Status הן Searching, Connecting, Downloading, Uploading ועוד).

השדה Seeds רלוונטי רק כאשר מתבצע Leeching והוא מציג את מספר ה-Peers המחזיקים בקובץ.

השדה Progress רלוונטי גם הוא רק כאשר מתבצע Leeching והוא מציג מספר בין 0 ל-100 המייצג את אחוז ההורדה אשר כבר התבצע.

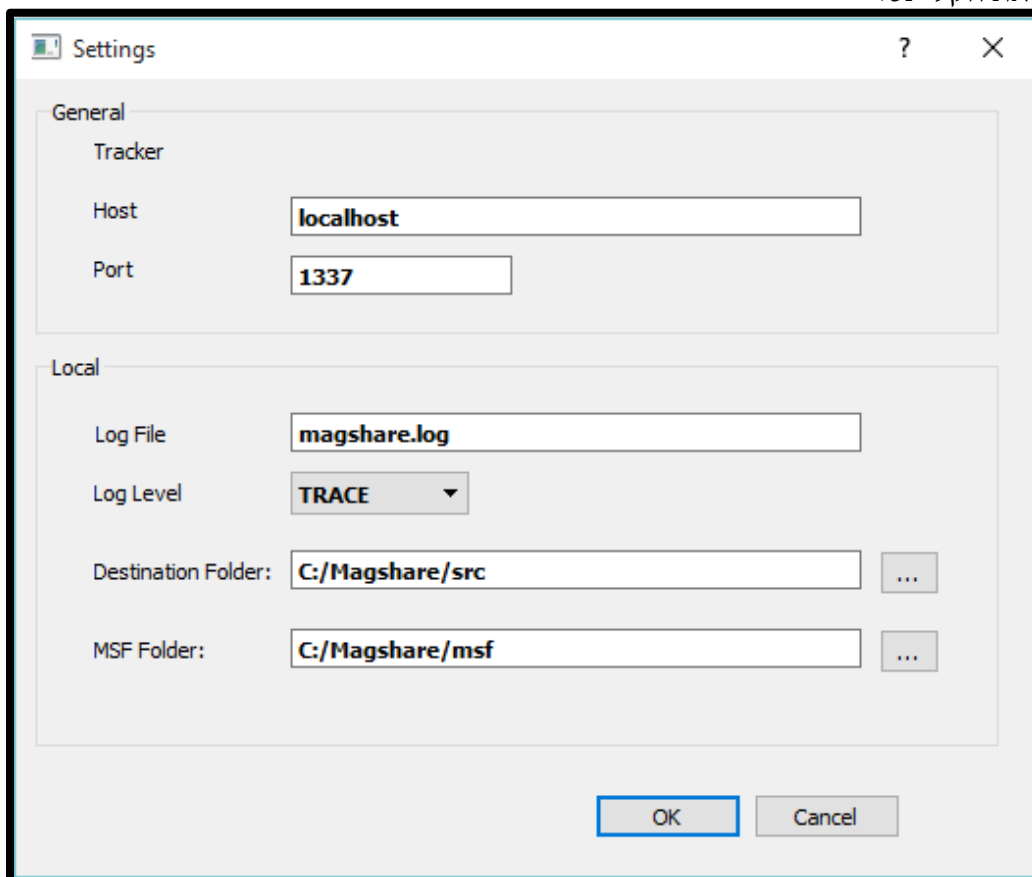


זהו המסך המתקבל כאשר לוחצים על File (מצד שמאל). Add MagFile יפתח חלון בו ניתן לבחור קבצים מסוג MSF, בכדי להתחיל Leeching או Seeding.

Settings יפתח חלון ובו הגדרות התכנה (מפורט בהמשך).

Create MagFile יפתח חלון ובו ניתן לבחור קובץ מכל סוג שהוא; לאחר בחירת הקובץ יפתח חלון נוסף בו ניתן יהיה לשמור את קובץ ה-MSF שנוצר מהקובץ הנבחר.

Exit סוגר את הקליינט.



זהו החלון שמופיע כאשר לוחצים על Settings מתוך התפריט File. בחלון זה ניתן להזין את הגדרות ה-Tracker (Host ו-Port), להגדיר את קובץ ה-log ואת רמת ה-logging (INFO, TRACE, DEBUG, WARN, ERROR, FATAL).



בנוסף ניתן לקבוע את תקיית היעד של הקבצים המיועדים ל-Seeding או היכן ישמרו הקבצים אותם מורידים וכן לקבוע את התקיה בה יישמרו קבצי ה-MSF.

בתפריט Help קיים מסך About המציג מידע על התכנה, מפתחיה וגרסתה הנוכחית.

## 6. רפלקציה אישית

באופן כללי, נהניתי מאוד בחצי השנה האחרונה בתהליך העבודה על הפרויקט. למדתי המון בנושא תקשורת וגם בנושאים אחרים (שימוש ב-Qt, הצפנות ונושאים רבים אחרים) ושיפרתי מאוד את הידע שלי בתכנות.

הקושי המרכזי עבורי היה עזיבתו של שותפי לפרויקט את הפרויקט עוד לפני כתיבת הקוד, מה שגרם לכך שכתבתי את כלל המערכת בעצמי מבלי שתתאפשר חלוקת עבודה בין שותפים לפרויקט. מצב זה גרם לי לעומס רב, אשר הקשה עלי לעיתים בעמידה בזמנים. כמו כן, התקשיתי לשמור על קשר מתמיד עם המנטור משום שעבדתי בשעות לא שגרתיות על הפרויקט (בעיקר מאוחר בערב ובסופי שבוע).

על אף הקשיים, הצלחתי לסיים את הפרויקט ואני שמח על כך שניתנה לי ההזדמנות ללמוד כה הרבה ולהעשיר את הידע שלי.

אחד מהקשיים במהלך הפיתוח היה יישום Multi-Threading, במיוחד בצד של הקליינט. הסיבה לכך היא שקשה מאוד לבצע Debug על מערכות שעובדות ב-Multi-Threading. הצלחתי לעקוף קושי זה באופן חלקי הודות למערכת ה-logging הנרחבת אשר שילבתי במערכת.

אחד מההישגים המשמעותיים בעבודה על הפרויקט היה הצלחתי להפעיל Clients שונים על אותה מכונה, מה שדרש חיבור לפורטים שונים בעלייתו של כל Client.

## 7. המוצר המוגמר אל מול התכנון הראשוני

להלן מספר הבדלים בין התכנון הראשוני של המערכת לבין המוצר המוגמר :

סיבה	מוצר מוגמר	תכנון ראשוני	
בשל אילוצי זמן לא הספקתי לממש הורדה במקביל של קבצים	ניתן להוריד קובץ אחד בלבד בכל פעם	תתאפשר הורדה של קבצים שונים במקביל	<b>הורדת קבצים במקביל</b>
שינוי נרחב בארכיטקטורת המערכת לעומת התכנון הראשוני	הקובץ מכיל שדות רבים (מפורט במסמך העיצוב) ולא מכיל כתובת Tracker	הקובץ יכיל בתוכו מספר id וכתובת Tracker בלבד	<b>תוכן קובץ ה-MSF</b>
HTTP התברר כפתרון מסובך ובעייתי מידי	התקשורת מתבצעת באמצעות sockets	בתחילה חשבתי לבצע תקשורת באמצעות HTTP	<b>תקשורת</b>
החלטתי לבנות UI פשוט עקב כך שהתכנה תומכת בהורדת קובץ אחד בכל פעם	שורה לקובץ בודד	טבלה המציגה קבצים מרובים	<b>UI</b>
בשל קשיים בביצוע Debug למערכת Multi-Threading שילבתי מערכת logging שמועילה במעקב אחר פעולות התכנה	בעל מערכת logging נרחבת	לא היה בתכנון	<b>Logging</b>

## 8. ביבליוגרפיה

להלן החומרים הביבליוגרפיים בהם השתמשתי במהלך העבודה על הפרויקט:

- [.https://wiki.theory.org/BitTorrentSpecification](https://wiki.theory.org/BitTorrentSpecification)
- [.https://en.wikipedia.org/wiki/BitTorrent](https://en.wikipedia.org/wiki/BitTorrent)
- [.https://github.com/victronenergy/QsLog](https://github.com/victronenergy/QsLog)
- [.https://en.wikipedia.org/wiki/SHA-1](https://en.wikipedia.org/wiki/SHA-1)
- [.http://doc.qt.io/qt-5/classes.html](http://doc.qt.io/qt-5/classes.html)
- [./http://www.kristenwidman.com/blog/33/how-to-write-a-bittorrent-client-part-1](http://www.kristenwidman.com/blog/33/how-to-write-a-bittorrent-client-part-1)