**ControlNet**

מערכת בקרה ושליטה במעבדת מחשבים

יובל כהן

**<גרסה ראשונה>**

**<8.2.2020>**

היסטוריית גרסאות המסמך

|  |  |  |
| --- | --- | --- |
| **תאריך** | גרסה | **תקציר השינויים** |
| 8.1.2020 | 1.0 | הוספת מסמך האיפיון - הקדמה. |
| 12.1.2020 | 1.1 | הוספת מסמך האיפיון - תיאור כללי, דרישות מפורטות על הפרוייקט, נספחים. |
| 8.2.2020 | 1.2 | הוספת מסמך העיצוב – התחלה. |
|  |  |  |

**קישור ל-github:** [**https://github.com/yuval130402/Final\_Project**](https://github.com/yuval130402/Final_Project)



תוכן עניינים

[הקדמה 3](#_Toc30003505)

[מטרה 3](#_Toc30003506)

[קהל היעד 3](#_Toc30003507)

[המוצר 3](#_Toc30003508)

[הגדרות חשובות להסבר המערכת 3](#_Toc30003509)

[תקציר 4](#_Toc30003510)

[תיאור כללי 5](#_Toc30003511)

[פונקציונליות המערכת 5](#_Toc30003512)

[אילוצים עיקריים 5](#_Toc30003513)

[הנחות ותלויות 5](#_Toc30003514)

[סביבת הפיתוח 5](#_Toc30003515)

[סיקור מצב השוק כיום 6](#_Toc30003516)

[דרישות מפורטות 7](#_Toc30003517)

[דרישות פונקציונליות 7](#_Toc30003518)

[דרישות של ממשקים חיצוניים 8](#_Toc30003519)

[דרישות לא פונקציונליות 8](#_Toc30003520)

[דרישות בסיס נתונים 8](#_Toc30003521)

[דרישות נוספות 9](#_Toc30003522)

[נספחים 14](#_Toc30003523)

[ממשק המשתמש (ה-GUI של המערכת) 14](#_Toc30003524)

[פיצ'ארים לשיפור (גרסאות פרוייקט) 15](#_Toc30003525)

# מסמך איפיון

# הקדמה

*מסמך זה מתאר את הפרוייקט שלי-* ***שליטה על מחשבים ובקרה עליהם ברשת מקומית (רשת LAN)****. הוא נועד לאפשר לקורא להבין מהו המוצר ומהי הפונקציונליות העיקרית שלו. במסמך איפיון זה ניתן למצוא הסברים על האופן בו פועל המוצר, על הפונקציות העיקריות בו ובנוסף הסבר למשתמש כיצד להשתמש במוצר.*

## מטרה

*מטרת מסמך האיפיון היא להנחות את המשתמשים כיצד להשתמש במוצר שלי. בנוסף, מטרתו להציג דרישות ופונקציות עיקריות, אילוצים והנחות של הפרוייקט.*

## קהל היעד

*המערכת מיועדת למורים/מרצים או לכל מנהל כיתת לימוד עם מעבדת מחשבים, המעוניינים במערכת שתאפשר בקרה, שליטה, וטיפול נוח במחשבים השונים בכיתה בזמן אמת, על מנת לאפשר מורה/מנהל להיות מודע למה שקורה במהלך השיעור ובמה התלמידים עסוקים במהלך השיעור. בזכות מערכת זו, המורה יוכל לשלוט במחשבי התלמידים ולפקח עליהם וכך ולהפוך את למידת התלמידים ליעילה ולאיכותית ככל הניתן.*

*קהל היעד צריך לדעת להפעיל קבצי פייתון ולאחר הפעלת תוכנת השרת במחשב שלו הוא צריך להריץ את תוכנות הלקוח במחשבי התלמידים כדי שיהיו מחוברים למערכת. המורה/מנהל הוא המשתמש ששולט על המחשבים המחוברים ויכול הפעיל פעולות שונות. קהל היעד צריך לוודא שכל ההנחות שאפרט בהמשך המסמך קיימות אצלו.*

## המוצר

* **שם המוצר: ControlNet .**
* ***תיאור המוצר:*** *המערכת תהיה מותקנת במחשבי הכיתה, ומחשב אחד (של המורה) יהיה המנהל והשרת, וכל מחשב של תלמיד הוא לקוח. המערכת מאפשרת למורה לצפות במסכי התלמידים, לנעול להם את המחשבים, לכבות ולהדליק את המחשבים מרחוק, שידור מסך המורה אל מסכי התלמידים, העברת קבצים ועוד פיצ'ארים נוספים שיקלו על המורה לשלוט במחשבי התלמידים. המחשב של המורה הוא היחיד שיכול לעשות את רוב הפעולות, כאשר לתלמידים יש פעולות מוגבלות כגון שליחת קבצים למורה על פי בקשתו.*
* ***מטרות המוצר ויתרונותיו:***

*מטרת המערכת היא לתת למורים מערכת שליטה ובקרה נוחה עם ממשק משתמש נוח לשימוש ולתפעול.*

*המטרה הכללית של המערכת היא לייעל את דרך הלימוד שקיימת כיום בבתי ספר ובהרבה מקומות לימוד, לאור ההתפתחות הטכנולוגיה בתקופתנו ולאור הצורך* ***בשינוי שיטת הלימוד הקיימת****.*

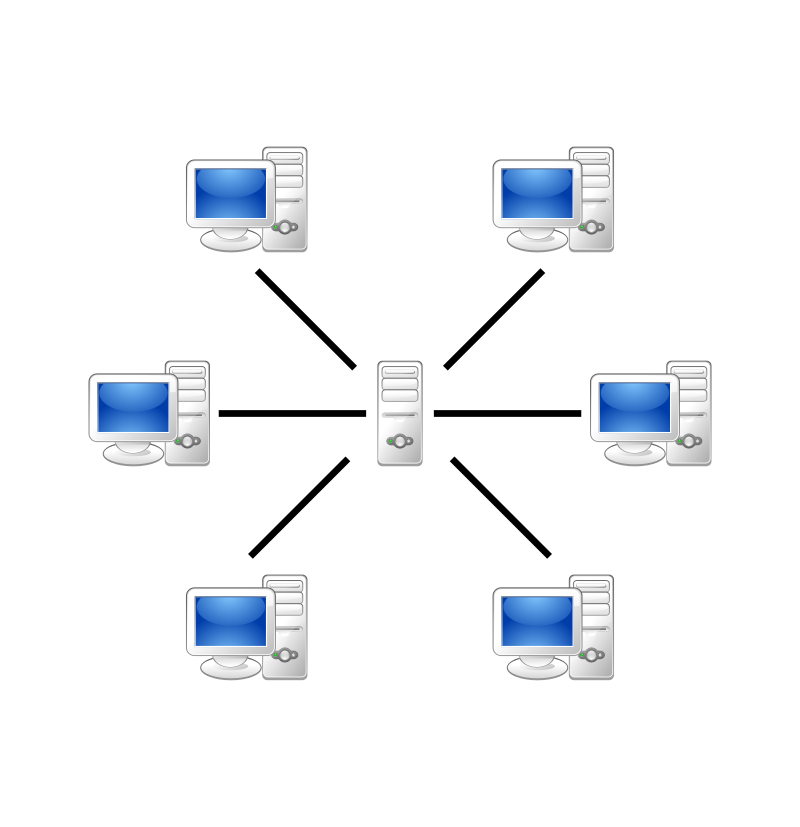
*לפי דעתי, שיטת הלימוד הקיימת בימינו פחות מקצועית ויעילה, בעקבות התפתחות הטכנולוגיה אני חושב שיש להעביר לפחות חלק משיעורי הלימוד מלמידה בכיתות לימוד (עם ספרים, מחברות ולוח למידה ) ללמידה בחדרי מחשבים ואז ההתקדמות הלימודית תהיה מהירה יותר ואופן הלימוד יהיה יותר יעיל ונוח.*

## הגדרות חשובות להסבר המערכת

* ***Python 3.7****- הגרסה של שפת התכנות python בה מימשתי את הפרוייקט.*
* ***Socket****- ממשק תוכנתי להעברת מידע בין תוכנות שונות. זהו API שמסופק בידי מערכת ההפעלה. Socket הוא נקודת קצה אחת של קשר דו-כיווני, של חיבור בין שני רכיבים ברשת.*
* **Window 10**- [*מערכת הפעלה*](https://he.wikipedia.org/wiki/%D7%9E%D7%A2%D7%A8%D7%9B%D7%AA_%D7%94%D7%A4%D7%A2%D7%9C%D7%94) *מסדרת*[*Windows*](https://he.wikipedia.org/wiki/Microsoft_Windows) *מבית* [*מיקרוסופט*](https://he.wikipedia.org/wiki/%D7%9E%D7%99%D7%A7%D7%A8%D7%95%D7%A1%D7%95%D7%A4%D7%98) *ל*[*מחשבים אישיים*](https://he.wikipedia.org/wiki/%D7%9E%D7%97%D7%A9%D7%91_%D7%90%D7%99%D7%A9%D7%99)*,* [*מחשבי לוח*](https://he.wikipedia.org/wiki/Microsoft_Surface)*,* [*טלפונים ניידים*](https://he.wikipedia.org/wiki/%D7%98%D7%9C%D7%A4%D7%95%D7%9F_%D7%A1%D7%9C%D7%95%D7%9C%D7%A8%D7%99) *וכו'.*
* ***Computer network*** *(רשת אינטרנט)- האינטרנט היא רשת תקשורת המאפשרת העברת נתונים והיא בעלת היקף כלל עולמי. הרשת נוצרה מחיבורים רבים בין רשתות מחשבים אשר איפשרו תקשורת בין מחשבים רבים ברשתות רבות.*
* ***LAN*** *(Local Area Network)- רשת מקומית. רשת מחשבים המתפרסת על אזור קטן ומוגבל, לרוב בתוך בניין או כיתה אחת.*
* ***Client****- מחשב לקוח המחובר למערכת.*
* ***Server****- מחשב השרת, שולט על הלקוחות ומספק להם שירותים או משאבים שונים.*
* ***Tkinter****- ספריית פייתון שבאמצעותה ניתן ליצור ממשק גרפי בפייתון.*
* ***PyCharm****- התוכנה בה עבדתי על קוד הפייתון ובה הרצתי את המערכת.*
* ***PAGE****-התוכנה בה יצרתי את ממשק המשתמש, את ה-GUI של המערכת. זוהי תוכנת עיצוב של ממשק משתמש, זה ה-designer של Tkinter בשפת python.*
* ***Sqlite3****-מודל של בסיס הנתונים שימוש ב-SQL,* צפייה במאגר הנתונים באמצעות התוכנה DB Browser (SQLite).

## תקציר

*בהמשך המסמך מופיע תיאור כללי של המוצר בתוכו מפורטים הפונקציונליות, קהל היעד אליו מכוון הפרוייקט, אילוצים שלקחתי בחשבון והנחיות. לאחר התיאור הכללי של מערכת השליטה ובקרת מחשבים, מפורטות דרישות המסבירות על המערכת: באמצעות דרישות אלו המשתמשים יכולים להבין כיצד מתנהלת המערכת.*



# תיאור כללי

החלק הזה מתאר את הגורמים העיקריים המשפיעים על המוצר ועל דרישותיו. חלק הזה לא מפרט דרישות ספציפיות אלא רק עוזר להבין את הדרישות.

## פונקציונליות המערכת

פעולות המורה/מנהל המערכת-

**המורה יכול להפעיל פעולה מסויימת אותה הוא רוצה לבצע על כל מחשבי התלמידים במערכת או על מחשב/מחשבים ספציפיים (כאשר הוא רוצה שהפעולה תתבצע** רק **על מי שבחר).**

**אופן פעולתו: בחירת המחשבים מתוך הרשימה שמופיעה לו ע"י סימון המחשבים הספציפיים, או לחיצה על כפתור** Select All Clients **הבוחר את כל המחשבים במערכת, ולאחר מכן לחיצה הפעולה הרצויה.**

**הפעולות:**

1. **צפייה במסך תלמיד מסוים שהוא בוחר מתוך התלמידים הקיימים במערכת.**
2. **שידור מסך המורה + עצירת השידור.**
3. **נעילת העכבר, המקלדת ומסכי התלמידים + שחרור נעילה זו.**
4. **כיבוי והדלקת מחשבים מרחוק.**
5. **שליחת קובץ לתלמידים.**
6. **צפייה ברשימת המחשבים הקיימים במערכת.**

**בהמשך אפרט על פעולות רקע (הקורות באופן אוטומטי).**

## אילוצים עיקריים

* המערכת דורשת חיבור לאינטרנט, המחשבים המתחברים למערכת חייבים להיות מחוברים לאינטרנט (לרשת המקומית).
* המערכת כתובה בשפת התכנות פייתון ואותה אני מריץ דרך תוכנת PyCharm. ביצוע התקשורת והחיבור בין מחשבים שונים (בין הלקוחות) לבין השרת ניעשה ע"י תקשורת שרת-לקוח, דרך שימוש ב-Socket.
* באופן תאורטי, המערכת מטפלת בכמות לא מוגבלת של לקוחות בו זמנית, אך דבר זה יבוא במחיר מהירות מחשב השרת וכמות העומס אותו הוא מסוגל להכיל. לכל צורך של מעבדת מחשבים גדולה (עד 40 מחשבים) לא יהיה עומס משמעותי.

## הנחות ותלויות

* *כל* המחשבים שמריצים את הפרוייקט (מחשבי מורה ותלמידים) מכילים Python 3.7 ומכילים את הספריות/חבילות בפייתון הדרושות להרצת התוכנה (פירטתי עליהן).
* המחשבים צריכים להכיל את מערכת ההפעלה Window 10.
* המחשבים צריכים להכיל את תוכנת devcon- כלי cmd שמציג מידע מפורט על התקנים במחשבים שבהם פועל windows. באמצעות devcon ניתן להפעיל, להשבית, להתקין, להגדיר ולהסיר התקנים של המחשב. צריך להוריד תוכנה זו דרך הקישור המצורף, והלכניס את קובץ התוכנה לספריית system32 שבwindows-.

<https://drive.google.com/file/d/0BwUy-0_C0imraVFjZUxtZTJXZEE/view>

* חובה שהמחשבים יהיו מחוברים באותה רשת אינטרנט (באותו ה-LAN) ורצוי להתקין את תוכנת PyCharm ולהריץ דרכה את המערכת.

## סביבת הפיתוח

Python 3.7- שפת התכנות.

Pycharm- כתיבת הקוד והרצה.

Sqlite3- יצירת מאגר הנתונים, DB Browser בשביל לצפות במאגר הנתונים.

Page ו-Tkinter- ליצירת ממשק המשתמש.

## סיקור מצב השוק כיום

ישנן מספר תוכנות המאפשרות שליטה ובקרה מרחוק, המפורסמות מבניהן הן VNC, NetSupport ו-Dameware המשמשות בתי ספר ומקומות עבודה כאחד. תוכנות אלו מאפשרות מספר רב של פעולות, אך הרבה פעמים הן אינן לגמרי הכרחיות למורים והשימוש בהן לא בהכרח נוח. התוכנה שלי מאפשרת את מרבית הפעולות השימושיות למורים/מנהלים, ועוד פעולות נוספות המונעות מתלמידים "לרמות את המערכת" (כגון ניתוק המחשב מהרשת), בעוד שמירה על שימוש מהיר ונוח.

# דרישות מפורטות

החלק העיקרי והחשוב ביותר של מסמך האפיון. פרק זה יכיל את הדרישות המפורטות מהמערכת שינחו את מעצבי המערכת, המפתחים, והבודקים בהמשך. כל דרישה שתפורט בחלק זה צריכה להיות:

## דרישות פונקציונליות

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | *דרישה* | *הסבר כללי על התהליך* | *קלט* | *פלט* | *טיפול בשגיאות* |
| 1. | שידור מסכי התלמידים למורה | *צילום מסך התלמיד הנבחר ע"י המורה והצגתו על מסך המורה.* | *בהעברת המסך מעבירים 2 דברים:*  *-גודל התמונה של מסך התלמיד (אורך הפיקסלים).*  *-תמונת מסך התלמיד (הפיקסלים עצמם).* | *הצגת מסך התלמיד על מחשב המורה.* | *אטפל בהמשך.* |
| *2.* | **שידור מסך המורה** | *צילום מסך המורה והצגתו על מסך התלמיד/מסכי התלמידים הנבחרים ע"י המורה. במקרה זה מסכי התלמידים הנבחרים, המקלדת והעכבר ננעלים והם יכולים לצפות רק בשידור מסך המורה.* | *בהעברת המסך מעבירים 2 דברים:*  *-גודל התמונה של מסך המורה (אורך הפיקסלים).*  *-תמונת מסך המורה (הפיקסלים עצמם).* | *הצגת מסך המורה על מחשב התלמיד.* | *אטפל בהמשך.* |
| *3.* | **נעילת העכבר, המקלדת ומסכי התלמידים + שחרור נעילה זו** | *התעלמות מכל קלט התלמיד ע"י נעילת המקלדת והעכבר ונעילה של המסך שלו באמצעות ראיית תמונה אחת.* | *מחשב התלמיד אותו נועלים.* | *הצגת תמונת נעילה כמסך מלא על מסך התלמיד והתעלמות מקלט התלמיד.* | *אטפל בהמשך.* |
| *4.* | **כיבוי והדלקת מחשבים מרחוק** | *מחשב השרת יכול לשלוט על מחשבי התלמידים ולהחליט מתי שהוא רוצה לכבות או להדליק אותם ע"י לחיצה על הכפתור המתאים.* | *מחשב התלמיד אותו מכבים או מדליקים.* | *כיבוי או הדלקת המחשב.* | *אטפל בהמשך.* |
| *5.* | **שליחת קובץ לתלמידים** | *העברת קובץ ממחשב המורה למחשב התלמיד או התלמידים הנבחרים.* | *הקובץ והמחשב אליו שולחים את הקובץ.* | *הקובץ יופיע בתיקייה מסויימת במחשב אליו הוא מועבר.* | *אטפל בהמשך.* |
| *6.* | **צפייה ברשימת המחשבים הקיימים במערכת** | *מוצג למורה בממשק המורה רשימת המחשבים המחוברים למערכת שעליה הוא שולט.* | *אין.* | *אין.* | *אין.* |

## דרישות של ממשקים חיצוניים

**3.2.1** ממשקי משתמש (למשל: GUI)- לכל משתמש (מנהל/מורה) מוצג ממשק גרפי ב- tkinter.

**3.2.2** ממשקי חומרה**- אין.**

**3.2.3** ממשקי תוכנה- עיצבתי את ממשק המשתמש באמצעות PAGE, תוכנה שיצרה לי באופן אוטומטי את ה-GUI של המוצר, יש צורך לתקשר בין קבצי ה-GUI לקבצי השרת לקוח, ובין קובץ השרת לקובץ הלקוח.

**3.2.4** ממשקי תקשורת**- תקשורת בין שר**ת ללקוחות ע"י Socket.

## דרישות לא פונקציונליות

*דרישות מהמערכת שלא מתבטאות בפיצ'ר ספציפי או בתהליך ספציפי שמתרחש במערכת אבל משפיעות על אופן עיצובה ומימושה, לדוגמא:*

**3.3.1 דרישות ביצועים (performance)**

**3.3.2 דרישות מהימנות (reliability)**

**3.3.3 דרישות זמינות (Availability)**

**3.3.4 דרישות אבטחה (security)**

**3.3.5 דרישות תחזוקה (maintainability)**

**3.3.6 דרישות ניידות (portability)**

## דרישות בסיס נתונים

* *במאגר הנתונים יישמרו נתונים על מחשבי התלמידים. השרת יוצר, כותב וקורא ממאגר נתונים במחשב המורה בשם Clients.db.*
* *יש לשמור את המידע כקובץ של מאגר נתונים עם סיומת .db*
* *רק לשרת יש גישה לבסיס הנתונים ויכולת לצפות בו.*
* ***שדות בסיס הנתונים:***

|  |  |  |
| --- | --- | --- |
| *שם השדה* | *טיפוס השדה* | *הסבר* |
| *ClientId* | *Integer, primary key* | *עולה אוטומטית, מספר סידורי של מחשב התלמיד. משמש כדי לשמור על בסיס נתונים מסודר ולנוחות.* |
| *Name* | *String* | *השם של מחשב התלמיד (מה שהתלמיד הגדיר).* |
| *IP* | *String* | *כתובת ה-IP של מחשב התלמיד, משמש לתקשורת.* |

## דרישות נוספות

*חבילות/ספריות מיוחדות שצריך להוריד לפייתון:*

* opencv, mss - צילום ועיבוד תמונה.
* zlib - דחיסת תמונה.
* socket - תקשורת.
* ctypes – שימוש בwinapi.
* pywin32.
* time - שימוש בזמן.
* pygame, tkinter - יצירת ה- GUI.
* threading - תהילכונים במקביל.
* pythoncom, pyWinhook - נעילת עכבר ומקלדת.
* Ctypes.
* Pynput – לעבודה עם העכבר והמקלדת.
* Pygame.

# מסמך עיצוב

ארכיטקטורת המערכת

חלק זה כולל את תיאור מבנה המערכת ופירוט המודולים השונים בה

2.1 מבט על

בסעיף זה יש להסביר את החלוקה העיקרית למודולים השונים במערכת, והיחסים/קשרים ביניהם. יש להגדיר באופן ברור את תחום האחריות של כל מודול או רכיב, ולאיזו פונקציונליות (כפי שהוגדרה במסמך האפיון שנכתב לפני כן) הוא מתקשר (בין אם מודול אחד בלבד או מספר מודולים יחד).  
חשוב להסביר לא רק מה/איך נעשה אלא גם מדוע דווקא כך.

2.2 פירוט רכיבי המערכת

**כאן יופיע פירוט לכל רכיב (ניתן לעשות באיטרציות – כל פעם עבור הרכיבים הרלוונטיים, לפני תהליך הפיתוח שלהם)**  
בסעיף זה יש לפרט את המבנה הפנימי של כל מודול/רכיב - כלומר:

* מבנה המחלקות השונות המרכיבות אותו, המאפיינים שלהם (public/private/protected + אילו טיפוסי משתנים כל פונקציה מקבלת ו/או מחזירה) והקשרים ביניהם (מי קורא למי, משתמש במי ולאיזה צורך) – מומלץ לצייר גרף UML (לא חייבים לפרט משתנים/פונקציות פרטיות אבל שיהיה ברור מה קשור למה ובאיזה אופן או לאיזה צורך)
* יש לתאר את ה- data flow / use case של שימושים שונים בכל תת-רכיב או פונקציה (מה קורה אם מתקבלים סוגים שונים של ערכים, איך/מדוע/באילו מקרים הרכיבים מתנהגים בצורה מסוימת או אחרת וכן הלאה)

**המלצות:**

* כדאי לתכנן רכיבים תשתיתיים עבור צרכים גלובליים במערכת - כמו למשל יצירת תקשורת רשתית, כתיבה וקריאה לבסיס נתונים (או קבצים), כתיבת תיעוד (log) הפעולות של המערכת (מקובל להגדיר "רמה" לכל שורה שנכתבת - debug, info, warning, error כדי שיהיה קל לסנן בזמן ריצת התוכנית. בנושא זה – מוזמנים להסתכל על מודול logging בפייתון).
* מומלץ לתכנן את הרכיבים באופן שיאפשר אבסטרקציה ומודולריות – בכדי לאפשר הרחבה עתידית ותחזוקה קלה. אם המימוש הפנימי של רכיב או מודול מסוים משתנה אז שאר הרכיבים/מודולים אינם מושפעים או נפגעים מכך ואין צורך לשנות גם אותם, או כאשר רוצים להוסיף פונקציונליות/רכיב חדש אז אין צורך לשכתב את כל שאר המערכת במיוחד בשבילו.

2.3 דיון בנושא העיצוב הנבחר

כאן יש לפרט ובעיקר להסביר מדוע בחרתם דווקא בחלוקת הרכיבים/תפקידים הזו ולא אחרת. מה היתרונות שלכם מבחינתה, וגם מה החסרונות שאתם מודעים אליהם. במידה וחשבתם על חלופות אחרות, יש לציין אותן ולהסביר בקצרה כיצד הן שונות מהפתרון שנבחר - ומדוע החלטתם בסופו של דבר שלא לבחור בחלופות האלו.  
כאן יש להתייחס גם לגבי שפת התכנות שבה בחרתם לכתוב כל רכיב, ומדוע דווקא בשפה זו ולא אחרת.

עיצוב נתונים ופרוטוקולים

כאן יופיע תיעוד של מבני נתונים שונים / פרוטוקולים המשמשים אותנו במערכת

בסעיף זה יש לפרט את כל סוגי המידע אשר מועבר או נשמר במערכת - בין אם בין רכיבים, בין מודולים או בכל צורה אחרת (כלומר גם אם זה משמש רק לתקשורת פנימית ולא ע"ג הרשת, או למשל אם זה משהו ששומרים לקובץ/בסיס נתונים).

* לכל סוג מידע כזה יש לפרט על השדות אותן הוא מכיל ומאיזה סוג כל שדה, מה טווח הערכים הרלוונטי לגביו וכל הגבלה או מידע נוסף אחר שאתם מוצאים לנכון (למשל: האם מותר שהשדה יהיה ריק ומה המשמעות של זה, אורך או ערך מינימלי/מקסימלי למספר או למחרוזת)
* יש לציין לאיזו מטרה משמש כל מבנה נתונים.
* עבור פרוטוקול תקשורת-נתונים בין רכיבים שונים:
* מה המצבים השונים לכל סוג של תקשורת (למשל אותנטיקציה, הרשמה, שליחת קובץ וכו')
* באילו ייצוגי מידע נעשה שימוש + תרשים זרימה של המצבים השונים שיכולים להיות וסדר השלבים שלהם (למשל כיצד נעביר מידע על מקרים של סיסמה שגויה, משתמש חסום, התחברות מוצלחת וכו')
* מומלץ להגדיר מראש קודים של בקשות/פעולות, תוצאות/תגובות ושגיאות אשר ישותפו בין כך הרכיבים הרלוונטיים
* יש להתייחס לאופי המידע הנשמר (האם הוא טקסטואלי או בינארי) ולייצגו בצורה נוחה והולמת בהתאם
* יש לפרט אודות מבנה בסיס הנתונים (חלוקה לטבלאות, השם והטיפוס של כל עמודה/שדה, קשרים בין טבלאות ואילוצים כלשהם על עמודות)

ממשק משתמש

כאן יופיע פירוט ממשקי המשתמשים וכן תופיע הסקיצה עבורם

בסעיף זה יש לפרט את הפונקציונליות של המערכת כפי שהיא מתבטאת עבור משתמש חיצוני. במידה ויש סוגים שונים של משתמשים, יש להתייחס לכולם בהתאם - אילו רכיבים או נתונים רלוונטיים עבורם, וכיצד הם מתקשרים איתם.  
  
בנוסף, יש לצרף המחשות ויזואליות של המסכים השונים, ולהסביר את התוכן שלהם (למשל מה התפקיד של כל שדה או כפתור, מה קשור/תלוי במה - למשל כפתור שמכובה בהתאם לתנאים מסוימים במערכת וכן הלאה) והקשרים ביניהם (איזה מסך מוביל לאיזה מסך ובאילו מקרים)  
  
דוגמה (לא מלאה):  

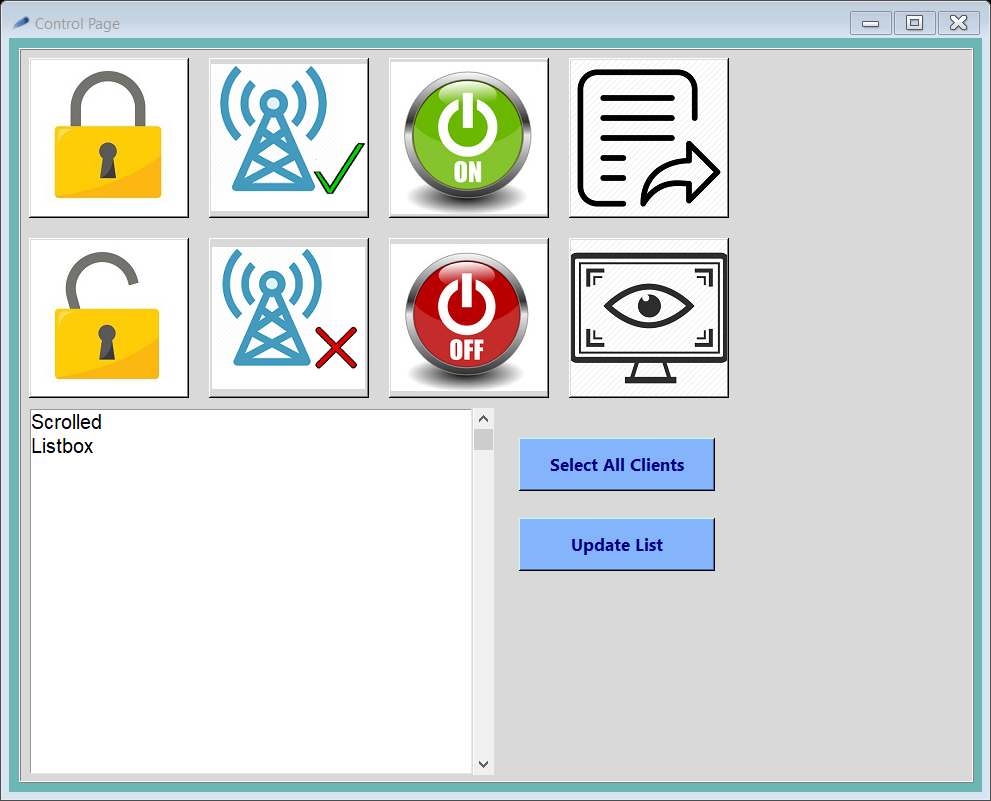

# נספחים

## ממשק המשתמש (ה-GUI של המערכת)

הדלקת מחשב

שידור מסך המורה

שליחת קובץ

****

עצירת השידור

בחירת כל המחשבים במערכת עליהם תבוצע הפעולה

נעילת העכבר, המקלדת ומסכי התלמידים

צפייה במסך המחשב הנבחר

שחרור נעילה

רשימת המחשבים הקיימים במערכת

עדכון רשימת התלמידים שמופיעה

כיבוי המחשב

## פיצ'ארים לשיפור (גרסאות פרוייקט)

|  |  |
| --- | --- |
| *גרסא ראשונה* | *גרסא שנייה* |
| *מערכת הכוללת נעילת מסך, שידור מסך, כיבוי והדלקת מחשב מרחוק.* | ***עדכון תוכנת התלמיד*** |
| *שליחת קבצים* | ***הוספת צ'אט*** |
| *אפשרות לבצע את הפעולה הנבחרת על מחשבים ספציפיים.* | ***התנתקות מהאינטרנט/כיבוי התוכנה, הפעלה אוטומטית.*** |