**שיתוף מסכים בכיתה**

מסמך אפיון

יובל מורן

גרסה 1

21.11.16

**היסטוריית גרסאות המסמך**

|  |  |  |
| --- | --- | --- |
| **תאריך** | גרסה | **תקציר השינויים** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |



**1. הקדמה**

**מטרת המוצר - לאפשר שיתוף מסכים בכיתה כך שמה שנמצא במחשב הראשי המחובר אל הלוח.המקרן יוכל (לפי רצון התלמידים והמורה) להיות משודר בכל המחשבים בעלי האפליקציה.   
קהל היעד של המוצר הוא תלמידים ומורים בכיתות ממוחשבות.**

**שם המוצר הוא ScreenShare. המוצר בעצם מאפשר שיתוף מסכים, כך שמה שמוקרן על הלוח הראשי יכול להיות מוקרן גם על כל אחד מהמחשבים בכיתה. על כל ילד שמעוניין לקבל הקרנה משותפת של המסך הרשי על מסך המחשב שלו להתקין את התוכנה על מחשבו האישי. ברגע שמתקיים שיעור והתלמיד רוצה להקרין את החומר על המחשב שלו, הוא נכנס לאפליקציה, בה הוא מוצא רשימה של כל המסכים המשדרים כרגע, הוא לוחץ על שם המסך של המחשב של המורה שלו ובכך מתחבר למסך.**

**מטרת המוצר היא לשפר את רמת הלמידה של תלמידים בעיקר בכיתות גדולות. כיתות לימוד יכולות להיות גדולות מאוד ולהכיל מעל ל40 תלמידים, בצורה כזו תלמידים שיושבים בחלקים האחוריים של הכיתה יכולים לפספס חלקים ממה שנכתב על הלוח או נלמד בכיתה רק בגלל מקום ישיבתם המרוחק. בצורה כזו המורה יוכל לשתף את החומר המוצג על הלוח לכל המחשבים בכיתה, כך שכל תלמיד שיביא מחשב יוכל לצפות בלוח מקרוב ובצורה הברורה ביותר.**

1.3 הגדרות

*בחלק זה יש לציין את כל ההגדרות, המושגים וראשי התיבות בהם תשתמשו בהמשך המסמך.*

**1.4** תקציר

*הסבר מה מופיע בהמשך המסמך ופירוט כיצד זה מסודר לאורך המסמך*

1. תיאור כללי

2.1 פונקציונליות *1. המורה פותח את תוכנת ScreenSharing במחשב המחובר למקרן.   
2. התלמידים פותחים את התוכנה במחשבים האישיים שלהם. בשלב הראשוני יהיה עליהם לבחור שם משתמש.   
3. המורה לוחץ על כפתור שהופך אותו לנראה לכל, הוא הופך ל"שרת".  
4. התלמידים פותחים במחשב שלהם רשימה של כל ה"שרתים" שניתן לבקש להתחבר אליהם.  
5. המורה בוחר את מי לאשר, מי שהוא אישר מקבל שיתוף של המסך שלו.   
6. פונקציה נוספת היא קבלת צילום מסך מהמחשב של תלמיד מבוקש למחשב של המורה ושיתוף התמונה לכל מחשבי התלמידים.*

2.2 קהל היעד

קהל היעד של המערכת הם התלמידים והמורים במערכת החינוך בחטיבה העליונה בלבד. (מסובך ולא נחוץ בביה"ס יסודיים). התוכנה הולכת להיות קלה לשימוש, מאוד אינטואיטיבית ולכן אין צורך מיוחד בידע טכנולוגי מעבר לידע בסיסי.

מורים - כל מורה שיש לו בכיתה מקרן ומחשב, ומשתמש במחשב לצורך הוראה. אין העדפה למורה במקצוע מסויים. דרוש יהיה רקע טכנולוגי בסיסי בכדי לתפעל את המערכת כמו שצריך, אך הכפתורים האינטואטיביים אמורים להפחית את הקושי בשימוש.  
תלמידים - כל תלמיד שמביא מחשב נייד לכיתה, ומעוניין לקבל שיתוף של המסך. ילמדו את הממשק בקלות.

2.3 אילוצים עיקריים   
המערכת דורשת חיבור לאינטרנט, ושכל המחשבים המשתמשים בה יהיו באותה רשת הוייפי.   
המערכת צריכה לתמוך במספר משתמשים רב.

2.4 הנחות ותלויות

המערכת תרוץ בסביבת ווינדוס. כל ילד המעוניין להשתמש בתוכנה יוריד אותה מראש על מחשבו האישי. לא יהיו בקשות לחיבור למחשב המורה ממחשבים מרוחקים\לא שייכים לכיתה.

**2.5** סיקור מצב השוק כיום (אופציונלי)

*במידה וידוע לכם על מוצרים קיימים כיום (אתרים, תוכנות, אפליקציות וכו') שנותנים פתרון דומה (גם אם חלקי) לרעיון שלכם - קשרו אליהם כאן או תארו אותם בקצרה. התמקדו באופן שבו הם פותרים או מתייחסים לסוגיות דומות לאלו שבחרתם, מה הפיצ'רים העיקריים שהם מציעים ובמה הם שונים מהמוצר שלכם.*

1. 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:

* מזוהה על ידי שם (או עדיף - מספר) באופן ייחודי
* מתועדפת (כלומר מה העדיפות שלה כלפי הדרישות האחרות)
* ניתנת לבדיקה
* נכונה ושלמה
* עקבית ולא מתנגשת עם דרישות אחרות

*שימו לב שישנם מספר סוגים של דרישות, כפי שמפורט בכותרות כאן:*

*יש להשקיע בחלק זה ולפרט את הדרישות בצורה ברורה ונוחה להבנה. יש לזכור שזה לא מסמך עיצוב- יש להמנע מתיאור מפורט מדי שמגביל את העיצוב הטכני (למשל – שפת תכנות) שיגיע בשלב מאוחר יותר של עבודתכם.*

*את הדרישות רצוי לחלק לקטגוריות הבאות:*

3.1 דרישות פונקציונליות

*תיאור של פיצ'רים ספציפיים במערכת. לכל דרישה פונקציונלית מומלץ לפרט:*

* *הסבר כללי*
* *קלט*
* *תהליך*
* *פלט*
* *טיפול בשגיאות*

*ניתן (ואף רצוי) להציג את הדרישות הפונקציונליות בפורמט של* [*תרחיש שימוש (use case)*](http://he.wikipedia.org/wiki/%D7%AA%D7%A8%D7%97%D7%99%D7%A9_%D7%A9%D7%99%D7%9E%D7%95%D7%A9)*. בכל "נקודת זמן" של תיאור התרחיש יש לפרט את כל האפשרויות השונות שעומדות בפני המשתמש או המערכת, ומה יכול לקרות בכל התפצלות שכזו (שימו לב להתייחס גם למקרים שליליים, כמו שגיאות או נתונים לא מתאימים, ולא להניח שהמשתמש תמיד פועל בדרך שלה אנו מצפים)*

3.2 דרישות של ממשקים חיצוניים

**3.2.1 ממשקי משתמש** (למשל: GUI)

**3.2.2 ממשקי חומרה**

**3.2.3 ממשקי תוכנה** (אם התכנה מחולקת לכמה תוכנות שונות שצריכות לתקשר בניהן)

**3.2.4 ממשקי תקשורת (למשל תקשורת בין שר**ת ללקוחות)

ממשק GUI - ממשק משתמש לתוכנה. אמורים להיות שני ממשקים שונים - אחד למורים ואחד לתלמידים.  
בממשק המורים יהיה כפתור אחד ורשימה אחת. הכפתור ייקרא VisibleToAll וכאשר המורה יילחץ עליו הוא יהפוך את עצמו ל"שרת" שמסוגל לשתף את המסך שלו לכל מי שיתחבר אליו. הרשימה תתעדכן כל כמה שניות ותכיל את כל התלמידים ("הלקוחות") שמבקשים להתחבר לשרת. ליד כל שם של תלמיד תהיה תיבה ריקה שניתן לסמן אותה בלחיצה עליה. ליד הרשימה יהיה כפתור שמאפשר סימון של כל הנמצאים ברשימה בבת אחת. לידו יהיה כפתור אישור שכאשר המורה יילחץ עליו כל התלמידים שהוא אישר להם להתחבר יתחילו לקבל שידור של המסך. בנוסף, למורים יהיה כפתור Student's screen shot, לחיצה עליו תפתח רשימה של כל התלמידים שמחוברים למורה, המורה יבחר תלמיד אחד ע"י לחיצה על השם שלו ואז על כפתור אישור, תישלח בקשה למחשב של התלמיד לצילום מסך וכשתחזור התשובה צילום המסך יופיע במחשב של המורה ובכך במחשב של כל מי שמחובר אליו.

בממשק התלמידים תהיה רשימה ServerList המתעדכנת כל כמה שניות ומכילה את כל המחשבים שניתן לבקש להתחבר אליהם כדי לקבל שיתוף של המסך שלהם. כאשר התלמיד ילחץ על אחד השרתים ולאחר מכן על כפתור אישור תישלח בקשה להתחבר לשרת. אם הבקשה אושרה ייפתח במחשב של התלמיד חלון המכיל את המסך של מחשב השרת, עם כפתור X ליציאה וחזרה לחלון הראשי, וכפתור הקטנה "-" למזעור המסך. אם הבקשה לא תאושר תופיע במחשב של התלמיד הודעה מתאימה.

ממשק תקשורת - שרת אחד מול לקוחות רבים - על השרת להיות מסוגל לתת מענה למספר רב של לקוחות, כאשר השימוש הכללי יתמקד בכיתות לימוד (שיכולות להכיל בין 20 ל40 תלמידים) אך צריכה להיות אפשרות גם לכיתות גדולות יותר.

3.3 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 ביצועים - live streaming מהיר ומתעדכן ללא הפסקה. תהליך התחברות מהיר בין הלקוחות והשרת.**

**דרישות זמינות - התוכנה צריכה להיות זמינה לכל מחשב (כרגע עם מערכת הפעלה ווינדוס)**

3.4 דרישות בסיס נתונים

* *איזה נתונים יישמרו*
* *באיזה פורמט יש לשמור את המידע (DB / קובץ . ניתן לציין גם כמה אופציות)*
* *מה יכולות האחסון הנדרשות (נפח אחסון לדוגמא)*
* *מה רמת האבטחה הנדרשת*

3.5 דרישות נוספות

*כאלה שלא ידעתם לאיזה חלק הם שייכות*

5. נספחים