

מערכת ניטור פעילות מחשבים
Computer Activity and Monitoring System
(CAMS)

יוני 2018



אולה נסиров
עדן רונן

מערכת ניטור פעילות מחשבים
Computer Activity and Monitoring System
(CAMS)



אולה נסиров
udy Ronen
מנחה אקדמי : ד"ר ארמין שמילוביץ'
מנחה מקצועי : מר רמי פלומבו
לקוח : מר מיכאל לובר, רמי'ד מחשב אישי, מערכת המחשב
אוניברסיטת בן גוריון

תוכן		
6.....	תקציר1
7.....	סקירת מנהלים2
9.....	מבוא3
9.....	הצורך במערכת4
9.....	תיאור המצב הנוכחי5
10.....	חולשות המערכת הנוכחיות והצורך במערכת חדשה6
11.....	מטרות המערכת המוצעת7
11.....	סקירה של מערכות קיימות	8.
11.....	LABSTATS8.1
11.....	Deep Freeze8.2
12.....	Kiwi Application monitor8.3.
12.....	סיכום מערכות קיימות8.4
13.....	מסקנות8.4.1
13.....	טכנולוגיות9
13.....	טכנולוגיות כללי9.1
14.....	ASP.NET MVC9.2
14.....	Django9.3
15.....	Rubi on Rails (RoR)9.4.
16.....	Web2py9.5.
16.....	סיכום טכנולוגיות9.6.
17.....	מסקנות9.6.1
18.....	בחירה בסיס נתונים10
18.....	Microsoft SQL Server10.1
18.....	Oracle10.2.
19.....	PostgreSQL10.3.
19.....	סיכום בסיסי נתונים10.4
20.....	מסקנות10.4.1
20.....	בחירה מתודולוגיות פיתוח11
20.....	מודל מפל המים11.1
21.....	Scrum11.2
21.....	Unified Process11.3.
22.....	סיכום מתודולוגיות פיתוח11.4
23.....	מסקנות	11.4.1.

23.....	סיכום חלק ראשון.....	.12
23.....	ניתוח ועיצוב המערכת.....	.13
24.....	אפיון ראשוני של המערכת.....	.14
24.....	דרישות.....	14.1.
24.....	עץ דרישות.....	14.1.1.
24.....	דרישות עיקריות.....	.14.1.2
27.....	ניתוח ועיצוב המערכת החדשה.....	.15
27.....	ארכיטקטורה.....	15.1.
28.....	תרכישי שימוש.....	.15.2
28.....	1-UC-צפיה בכיתת מחשב.....	15.12.1.
29.....	2-UC-הוספת מחשב לכיתת מחשב.....	15.12.2.
29.....	3-UC-שינוי מיקום מחשב בכיתת מחשב.....	15.12.3.
29.....	4-UC-מחיקת מחשב מכיתת מחשב.....	15.12.4.
30.....	5-UC-הוספת כיתת מחשב.....	15.12.5.
30.....	6-UC-מחיקת כיתת מחשב.....	15.12.6.
30.....	7-UC-יצירת דוח.....	15.12.7.
31.....	8-UC-תצוגת התראות.....	15.12.8.
31.....	9-UC-ניהול מדיניות התראות.....	15.12.9.
31.....	10-UC-הוספת הרשאת גישה.....	15.12.10.
32.....	11-UC-הורדת הרשאת גישה.....	15.12.11.
32.....	UML.....	.15.3
33.....	טרנסקציות ותרשיימי DFD.....	.15.4
33.....	DFD-0.....	.15.4.1
34.....	איסוף נתונים.....	.15.4.2
35.....	ניהול דוחות.....	.15.4.3
36.....	ניהול מפות.....	.15.4.4
37.....	ניהול התראות.....	.15.4.5
38.....	ניהול הרשאות.....	.15.4.6
39.....	עץ תפריטים.....	.15.5
39.....	סיכום עבודות ניתוח ועיצוב.....	.16
40.....	IMPLEMENTATION.....	.17
40.....	התוכנה.....	.18
40.....	בסיס הנתונים.....	.18.1
44.....	תהליך איסוף נתונים.....	.18.2
45.....	תרכישי שימוש.....	.18.3

45.....	צפיה בכיתות מחשב-1 UC-1	.18.3.1
47.....	עריכת כיתת מחשב UC-2,3,4	.18.3.2
50.....	הוספת כיתת מחשב UC-5	.18.3.3
51.....	מחיקת כיתת מחשב UC-6	.18.3.4
52.....	יצירת דוח UC-7	.18.3.5
58.....	תצוגת התראות UC-8	.18.3.6
59.....	ניהול מדיניות התראות UC-9	.18.3.7
60.....	הוספה הרשות UC-10	.18.3.8
61.....	הורדת הרשאה UC-11	.18.3.9
62.....	אבטחת מידע	18.4.
63.....	בדיקות	18.5.
66.....	ניסויים	.18.6
66.....	ניסוי בדיקת תהליכי הפקת דוחות	.18.6.1
66.....	ניסוי בדיקת תקינות איסוף הנתונים ותצוגת תפוצה נוכחית	.18.6.2
67.....	התאוששות מתקלות	.18.7
67.....	מבנה התיקיות והקובציים	18.8.
71.....	תצוגת המערכת	19.
71.....	דף התחברות למערכת	.19.1
72.....	דף ראשי לסטודנטים	.19.2.
72.....	דף צפיה בכיתת מחשב	.19.3.
73.....	דף עריכת כיתת מחשב	.19.4.
73.....	דף הוספה כיתת מחשב	.19.5
74.....	דף הפקת דוח	.19.6
74.....	דף דוח ממוצע שעות פעילות	.19.7
75.....	דף דוח פעילות מחשבים לפי שעות	.19.8
75.....	דף התראות ועריכת התראות	.19.9
76.....	דף הרשות, עריכת הרשאה והוספה הרשאה	.19.10
77.....	הטמעת המערכת	.20
77.....	הקמת תשתיות	.20.1
78.....	תחזוקה	.20.2
78.....	ארגוני	21.
79.....	סיכום	.22
81.....	נספחים	.23
81.....	מערכת תפוצה נוכחית	.23.1
81.....	תצוגת אתר	.23.1.1

81.....	תצוגת מפת כיתה23.1.2
82.....	מערכת נתוני תפוסת כיתות ציבוריות	23.2.
82.....	מסך כניסה למערכת23.2.1
82.....	טבלת ניצול מחשבים23.2.2
82.....	נתוני תפוסת כיתה23.2.3
83.....	LABSTATS23.3
83.....	ההיסטוריה שימוש23.3.1
83.....	שימוש בתוכנות23.3.2
84.....	מקסימום שימוש23.3.3
84.....	שימוש ממוצע23.3.4
85.....	מפה של מעבדה23.3.5
85.....	Deep Freeze23.4
85.....	מספר משתמשים במחשבים לפי תאריך	23.4.1.
85.....	שימושיות רישונות23.4.2
86.....	Kiwi Application monitor23.5
86.....	רשימת היישומים הנמצאים במעקב23.5.1.
86.....	שימוש ממוצע בזיכרון לפי יישום23.5.2
86.....	רשימת מחשבים תחת פיקוח המנוי kiwi admin23.5.3
87.....	הודעות שהועברו לkiwi admin מחשב תחת פיקוח23.5.4
88.....	Summary	24.
89.....	Executive Review	25.

איורים

24	אייר 1 - עץ דרישות
27	אייר 2 - ארכיטקטורה
28	אייר 3 - תרחישי שימוש
32	אייר 4 - תרשימים מחלקות
33	אייר 5 - תרשימים זרימת מידע ראשי
34	אייר 6 - איסוף נתונים
35	אייר 7 - הפktת דוחות
36	אייר 8- DFD הצוגה וערכה של מפות כיתות מחשבים
37	אייר 9 - DFD הצוגה והגדלה של התראות
38	אייר 10 DFD ניהול הרשות
39	אייר 11 - עץ תפריטים
41	אייר 12 - סכמת בסיס הנתונים

טבלאות

12	טבלה 1 - מערכות קיימות
17	טבלה 2 - טכנולוגיות
19	טבלה 3 - בסיסי נתונים

22	טבלה 4 - מתודולוגיות פיתוח
40	טבלה 5 - פירוט גרסאות
41	טבלה 6 - טבלת Activities
42	טבלה 7 - טבלת Computers
42	טבלה 8 - טבלת Labs
43	טבלה 9 - טבלת ComputerLabs
43	טבלה 10 - טבלת Departments
43	טבלה 11 - UserDepartments
43	טבלה 12 - טבלת Users
66	טבלה 13 - בדיקות ייחודית
68	טבלה 14 - מבנה תיוקנית View
69	טבלה 15 - מבנה תיוקנית Scripts
70	טבלה 16 - מבנה תיוקנית Models
70	טבלה 17 - מבנה תיוקנית Images
71	טבלה 18 - מבנה תיוקנית Controllers

1. תקציר

פיתחנו מערכת המציגת תפוצה נוכחת והיסטוריה של מחשבים בכיוון המחשב הציבורי באוניברסיטה בן גוריון. המערכת מאפשרת להפיק דוחות סטטיסטיים על פעילותם של מחשבים.

דרישות המערכת אופיינו על ידי מר מיכאל לובר, ראש מדור מחשב אישי במערך המחשב של האוניברסיטה. המערכת שמורה לנוטנה לכל דרישות הלכה וצריכה כפי שאופיינו. בתכורתה המערכת נוחה לשימוש, ברורה ומעוצבת. המערכת אוספת מרוחק את נתוני השימוש במחשבים הפזוריים באוניברסיטה. המערכת משתמש בפועל את כל אחראי המחשב בפקולטות השונות, מנהליים ומקבלי החלטות בדרגות בכירות באוניברסיטה. כמו כן תשמש בפועל את מקבלי החלטות בהבנה عمוקה יותר על השימוש במחשבים ותעזר בקבלת החלטות הקשורות בהוספה של מחשבים וכיותם חשוב חדש או צמצום של מספר מחשבים והפיכת כל כיותם חשוב לכיתת לימוד וגילות וצדומה. המערכת תאפשר גם לסטודנטים למצוא מחשבים פנויים ברחבי הקמפוס.

המערכת מומשה בטכנולוגיות מתקדמות. המשק הניהולי וממשק משתמש הקצה הם אינטראקטיביים. ניתן לצפות בתपוטת המחשבים בזמן אמיתי, לעזרך את כיותם המחשב בקלות ולהפיק מגוון דוחות שימושיים המנתחים את פעילותם המחשבים באוניברסיטה ולהציגם בצורה בהירה וגרפית.

במהלך ביצוע הפרויקט השתמשנו בכלים שלמדו במהלך התואר כגון: אפיון מערכת, הגדרת דרישות, ניתוח ועיצוב מערכת, פיתוח בסביבה אינטראקטיבית, תקשורת ואבטחת מידע, עבודה עם בסיסי נתונים, בדיקות תוכנה, ניהול פרויקט ועוד.

המערכת הוטמעה למעשה בראשת האוניברסיטה בסביבה המדמה את מבנה הרשות ומມתיינה לאישור לפעולות שוטפות באתר האוניברסיטה.

למערכת מספר מגבלות: א. מיקום בתחום האוניברסיטה, על שירותי הקיימים באוניברסיטה. ב. מספקת שירות בעברית בלבד. ג. לא מותאמת לטלפונים חכמים.

לאורך הפרויקט התמודדנו עם אתגרים בנושאים שונים ומגוונים כמו: רשות, לצפות את כל התרחישים האפשריים ומרקם הקצה שלהם, עבודה בצוות, ניהול זמנים, עבודה מול לקוחות, עיצוב מערכת (פשיטת ובהירות, אינטואטיביות).

לא ספק, ביצוע פרויקט זה תרם לנו מאוד בכך ששימש אינטגרטור של ידע ויכולות שצברנו ובביצוע הפרויקט יכולנו להביאם לכל מעשה.

2. סקירת מנהליים

המחשבים הציבוריים באוניברסיטה הם משאב יקר הן מבחינת כוח אדם והן מבחינת ציוד. קבלת החלטות על משאבי אלו כגון הוספה של מחשבים, הוספה של כיתות מחשב, חלוקת משאבי וצדומה מתבצעת בהתחשבות בכמות הסטודנטים הנמצאים בפקולטה, שיעורים שבהם משתמשים ביישומי המחשב וגורמים נוספים המשפיעים על שיקולים אלו.

בעקבות שינויים טכנולוגיים, בעלות יותר ויוטר סטודנטים מחשבים ניידים, שירותים הענן וצורת השימוש בתוכנות מאפשרת לסטודנטים שימוש בתוכנות לבדות ויקרות במחשבים האישיים שלהם. כתוצאה לכך השימוש במחשבים הציבוריים ובכיתות המחשב באוניברסיטה הולך ופוחת. למורות שינויים אלה, עדין יש צורך רב במחשבים הציבוריים הום לצורך תרגול ועבודות הסטודנטים והן בהעברת שיעורים פורטניים. כיום ישנים כ-2000 מחשבים ציבוריים באחוריות האוניברסיטה. לצורך ניצול עילית יותר של המשאבי יש צורך לעקוב אחרי השימוש במחשבים לקבלת תמונה רחבה וברורה יותר.

הגורם האחראי על ניהול ותחזוקה של המחשבים הציבוריים הוא מערך המחשב של האוניברסיטה. ביום קיימות שתי תוכנות שונות אשר עוסקות בנושא המחשבים הציבוריים. מערכת "תפוסות כיתות ציבוריות" אשר מציגה את השיעורים המתוכננים בכל יום ואת מצב התפוסה של כיתות המחשב באוניברסיטה, ומערכת "נתוני כיתות ציבוריות" שמאפשרת תצוגה של ממוצע שעوت פעילות של מחשבים ציבוריים לפי כיתה ולפי תחנת קצה, בטוחה זמינים מוגדר. מערכות אלו ישנות ובעלות מגבלות רבות כמו חוסר תאימות למערכות חדשות, זמני תגובה איטיים ועובדת עם קבצי טקסט מנופחים, ממשקים יתנים ובאגים. במהלך השנים, כשלעצמה המודעות לתקלות אלו, היו ניסיונות לתקן ולשפר את הקוד המקורי אך השפה והטכנולוגיות בה נכתבו הפכו מושננות ולא תמכו בטכנולוגיות חדשות. כמו כן כל "יד" שנכנסה לקוד שינה אותו והוא נהייה מסורבל ולא מובן. ביום המערכות כבר לא מפיקות מידע אמיתי ולא ניתן להשתמש בהן כמעט כלל.

הפתרון שלנו הינו **מערכת ניטור פעילות מחשבים** אשר משתמשת בטכנולוגיות חדישות ומשלבת בתוכה את תצוגת המחשבים ותפוסתם ברוחבי הקמפוס ומערכת ניהול שמאפשרת ניהול פשוט של תצוגת המחשבים ובנוסף הפקט דוחות שימושים על פעילותם של המחשבים הציבוריים.

בתחילת הפרויקט הבנו וחקרנו את המצב הנוכחי ואת הבעיה אימה אנו מתמודדות, נפגשנו עם מר מיכאל לובר, ראש מדור מחשב אישי במערך המחשב של האוניברסיטה, הבנו ממנו את היצ妃ות מהמערכת, את האיליצים והמגבליות. הגדרנו את צרכי המערכת ובחנו חלופות אפשריות. מסקירת המערכות הקיימות והשוatan ראיינו כי/non לא מספקות את הצרכים הבסיסיים של המערכת הנדרשת, הגיענו למסקנה שהצעד הנכון יהיה לפתח את **מערכת ניטור פעילות מחשבים**.

לאחר מכון סקרנו טכנולוגיות ומתקדמויות פיתוח אפשריות כדי למש את המערכת. בחרנו להשתמש בסביבה עבודה ASP.NET MVC מכיוון שהיא עמדת ברוב הקורסרים שלנו שגדרכנו ובנוסף יש לנו הכי הרבה ניסיון אותה. בסיס נתונים Microsoft SQL כי הוא אידilli לעובדה עם

סביבת העבודה שבחרנו וגם עם המערכת הזו יש לנו הכוורת מעמיקה. למתודולוגיות פיתוח בחרנו את Unified Process לא הייתה לנו הכרות מוקדמת עם מתודולוגיה זו אך אחראי בחינת מתודולוגיות שונות מצאנו כי מתודליק הפיתוח מתחילה כאשר המטרות ברורות ומוגדרות היטב, בנוסף מתחילה החלקים המרכזיים והקריטיים ביותר למערכת ומשיכים בפיתוח לפי תרחישי שימוש מה שמאוד מתאים לאופי הפרויקט שלנו.

בשלב הבא ניתחנו ועיצבנו את המערכת המשודרגת, הגדרנו דרישות ואת הפונקציות אותן היא תכלול בדגש על פיתוח מערכת נוחה, ייעילה ושימושית. לצורך הבנת מבנה המערכת העתידית יצרנו תרחישי שימוש, תרשימי זרימת מידע (DFD) ותרשים מחלקות. שלב זה הינו משמעותי במחזור חיי המערכת ועל כן השקענו בו מחשבה רבה. נעשה ניתוח קפדי על מנת לצמצם ככל האפשר אי הבנות ותקלות שיכולות לקרוות בעקבות כך בשלב מאוחר יותר.

בסיום שלב ניתוח ועיצוב המערכת ויגשו למימוש התוכנה. את הפיתוח חילקו לאייטרא齊ות, כך שבכל אייטרא齊ה מישנו תרחיש שימוש אחר בהשראת מתודולוגיית הפיתוח שבחרנו. הפיתוח בארכיטקטורת MVC אפשר לנו חלוקה ברורה ונוחה של העבודה כך שאחת התרכזה בפיתוח שכבת התצוגה והשנייה בפיתוח שכבת המודול. עבודה בצדקה זאת מאפשרת לנו ריכוז ומיוקד.

לאחר פיתוחה של רוב המערכת שבנו למר מיכאל לובר על מנת שיבחנו את המערכת ויראה האם עד כה היא עונה על צרכיו. מר לובר אהב את המערכת ונתן דגשים ודרישות נוספות. בנוסף, בשביל תהליך הבדיקות וההטמעה הקצה עבורנו משאבים מטעמו לצורך בנית סביבת בדיקות כמה שייותר דומה לסייעת הייצור בתוך רשות האוניברסיטה.

לאחר שהתקנו את המדרשות הדדרשות, הגדרנו והתאמנו את המשאבים לצרכי המערכת שפיתחנו החילנו בבדיקות מקיפות. בדקנו שכל חלקי המערכת עובדים ייחדיו ובחנו את אמינותן איסוף הנתונים ותוצאות החישובים הסטטיסטיים של המערכת.

כיום מערכת ניטור פעילות מחשבים היא מערכת שלמה אשר נותנת מענה לכל דרישות הלקוח וצרcioו כפי שאופיינו. המערכת אוספת מרחוק את נתוני השימוש במחשבים הפזרים באוניברסיטה. הממשק ניהול וממשק משתמש הקצה הם אינטראקטיביים. בתצורתה המערכת נוחה לשימוש, ברורה ומעוצבת. ניתן לצפות בתფוסת המחשבים בזמן אמיתי, לעורוך את CITOTUT המחשב בקלות ולהפיק מגוון דוחות שימושיים המנתחים את פעילות המחשבים באוניברסיטה ולהציגם בצדקה בהירה וגרפית.

במשך **מערכת ניטור פעילות מחשבים** תוכג מול גורמים מנהליים באוניברסיטה לאישור ולפעולת שוטפת באתר האוניברסיטה.

3. מבוא

מחלקת המחשב מופקדת על פיתוח ותחזוקת מערך תשתיות המחשב והתקורת לצרכי המחקר, ההוראה ומערכות המידע המנהליות באוניברסיטה. מנהל מערך המחשב ממונה על המדריכים: מערכות הפעלה מרכזיות, מחשב אישי, תקשורת והפעלה, תמיינת מחשב, טכנולוגיות למידה ויעץ. **מדור מחשב אישי** מופקד על בחירה והטמעה של טכנולוגיות בתחומי המחשב האישי. המדור נותן תמיכה בתוכנה ובחומרה, כולל הדרכה בתחומים אלו. כמו כן עסק המדור בתכנון, הקמה ותחזוקה של כיתות המחשב הציבוריות להוראה ברחבי האוניברסיטה.

מסמך זה מסכם את השלב הראשון בעבודת פרויקט הגמר שמהוותה הקמת מערכת סטטיסטית למשך ודוח על פעילות המחשבים באוניברסיטה. השלב הראשון הוא לימוד המצב הנוכחי, הבנת הצללים וסקירת כלים מתאימים. השלב השני יכלול הכנת מפרט הנדי והשלב השלישי מיימוש התוכנה. השלב הרביעי יכלול ניסויים שילוב והרצה.

4. הצורך במערכת

באוניברסיטה בן גוריון נמצאות עשרות כיתות מחשב ציבוריות אשר משמשות את ציבור הסטודנטים לצרכי למידה עצמית והוראה. מרבית כיתות המחשב הציבוריות פתוחות 24 שעות ביממה, 7 ימים בשבוע, 364 ימים בשנה. לפני יותר מעשור כאשר השימוש במחשבים הציבוריים באוניברסיטה היה רחב מאוד סטודנטים עברו בין האתרים השונים במטרה למצוא מחשב פנוי לצרכי למידה. עליה צורך לאפשר לסטודנטים לאתר מראש מחשבים פנויים ובכך לחסוך זמן כיתות רגילים ותסכול מהנהליק.

קיים צורך מתמיד בבדיקה נצילות המחשבים בכיתות לצרכי ניהול. לדוגמה - אם הقيתה בניצול שנתי של פתוח מ-5% אולי לא ראוי לשמר על הقيתה זו ואין צורך המשיך להחזיק את כל המחשבים בכיתה וניתן להמיר כיתה זו לכיתת לימוד פרונטאלית (לא מחשבים) או משרדים.

עלות המחשבים ותחזוקתם היא גבוהה וכן יש צורך לוודא את מידת השימוש/מספר המחשבים הדרושים.

5. תיאור המצב הנוכחי

כיום קיימות 2 מערכות נפרדות, האחת للتצוגת מפות והשנייה לאיסוף שעות פעילות. במערכת הראשונה קיים תהליך (Process) אשר מותקן בכל המחשבים שככל פרק זמן של שלוש דקות שולח עדכון לשרת וכך המערכת יודעת איזה מחשב מחובר, כך שככל שלוש דקות המערכת יוצרת דף PHP בשרת, וזה מוצג באתר (נספח 1223.1). את המפה בונים بذلك שרת שכותב ב-*Visual Basic 5* ורץ על *Microsoft Server 2003*. המערכת שומרת את הנתונים על פרישת המחשבים לכיתות ופיזורים בכיתות, במאגר נתונים *SQL Server*. את סידור הכתובות והחלוקת לכיתות עשוה מושגתה ממערך המחשב בעזרת השרת ע"י ממשק מוגבל. למשל כתיבת נקודות במרחב ידנית (X,Y) שמייצגות את מיקום המחשב, אם רוצים

לשנות את מיקום כל המחשבים ולעשות הזחה של מספר מחשבים צריך לעשות זאת בכל מחשב ע"י הזנה מחדש של נקודות הצוין במטה. עקב המוגבלות, גם אם קיימים שינויים, מנעים מלעדכן ולהשתמש以此.

המערכת השנייה, מערכת "נתוני כיתות ציבוריות", משתמש בשירות (Service) שモתקן בכל המחשבים אשר שומר קבוע לוג עם שעות הפעולות של המחשב ובסוף כל יום השירות מעביר את הקובץ שעדכן במהלך היום לתיקיה משותפת בשרת לתוכן קובץ טקסט שמייצג את המחשב. הנתונים מוצגים במכשיר משתחם שבו ניתן לבחור תאריכים ושבועות וראות את שעות הפעולות הממצאות לפי כיתה וכן לפי תקופה (נספח 23.2).

6. חולשות המערכת הקיימת והצריך במערכת חדשה

- א. שתי המערכות{k}קיימות מתעדות את זמני הפעולות של המחשבים ולכן אין צורך בשתי מערכות שונות לשם כך.
- ב. המערכות ישנות ולא עובדות באופן מושלם עם הטכנולוגיות החדשות:
 - איסוף נתוני החתבות לא עובד כשרה עם Microsoft Windows חדש יותר Windows 7.
 - תצוגת מפות המחשבים בד"כ לא עובדת ואם כן לא ניתן תמונה נcona של המחשבים (מי בשימוש או לא בשימוש וכן אם אין תקשורת עם המחשב).
 - השירותים עובדים על שירות Microsoft Server 2003 ועליהם כך מערך המחשב הקיים מכונה וירטואלית עם Microsoft Server 2003 ועל המכונה השירות של המערכת רץ. המערכת צריכה להתאים לטכנולוגיות החדשות ולהציג נתונים אמינים.
- ג. כמות קבצי הטקסט שנשמרים בשרת במערכת "נתוני כיתות ציבוריות" היא גדולה מאוד (1800 קבצים) והרבה קבצים אינם רלוונטיים כבר כי חלק מהמחשבים הושבתו ולא קיימים יותר. הקבצים גדולים בנפח (4GB) וגדלים מדי יום, הדבר תופס מקום וגם מעmis על זמן פתיחת ושליפת הנתונים ממנו.
- ד. המערכת מציגה סוג אחד של דוח 매우 ספציפי של שעות שימוש לפי כיתות, יש צורך בהוספה סוג דוחות, אפשרויות סינון ומילוי המידע ותצוגה גרפית ועוד בהירה של הנתונים.
- ה. מרביתם של המחשבים הציבוריים מבוססים על מערכת הפעלה של Microsoft אך חלק מהכיתות מכילות מחשבים עם מערכת הפעלה Linux, המערכות כיוון לא עובדות עם מערכות הפעלה שונות מMicrosoft וכן אין מעקב אחרי מחשבים אלו. יש צורך שהמערכת תאוסף נתונים גם של מחשבים אלו.
- ו. ממשקי המשמש מואוד יישנים, לא נוחים לשימוש ולא יפים. יש צורך ממשק נייחולי איטראטיבי, מודרני ונוח לשימוש.
- ז. הקוד המקורי מסובך להבנה וכן השפה בה הוא רשום ישנה. במהלך השנים ניסו במערך המחשב לבצע שדרוג אך הניסיונות לא צלחו. נראה כי יש צורך במערכת אחת מודרנית, עדכנית ונוחה.

7. מטרות המערכת המוצעת

- .א. בדיקת נצלות מחשבים הנמצאים בכיתות המחשב הציבוריות באוניברסיטה.
- .ב. הצגת זמינות המחשבים הנמצאים בכיתות המחשב הציבוריות באוניברסיטה.
- .ג. מעקב ודיווח של פעילות מחשבים ציבוריים של האוניברסיטה.
- .ד. יצירה של סטטיסטיות על הנתונים והפרמטרים הנאספים.
- .ה. לאפשר הצגה וצפיה בתונתי השימוש והפקת דוחות שימושיים.
- .ו. לאפשר שימוש יעיל במשאים להקטנת ההוצאות.

8. סקירה של מערכות קיימות

LABSTATS .8.1

התוכנה מספקת פתרון מבוסס ען למדידת השימוש בטכנולוגיה במעבדות מחשבים. דוחות הנחוצים להסתדרות אינטראקטיבית ומאפשרים לראות נתונים נטוני שימוש במכשיר ובקלות. במערכת תצוגה של היסטוריות נתונים שימוש (נספח 23.3.1), כגון שימוש בתחנות כצה לפיקובצה, תגי תחנה או קבוצת תגים, פעילות המתרחשת בנקודת זמן ספציפית, דוחות מסודרים ממויינים לפי מאפיינים שונים וכן נטוני הפעולות של מחשבים וזמן שימוש. בנוסף, המערכת מציגה דוחות שימוש בתוכנות (נספח 23.3.2), תצוגה שללוחות זמינים היסטוריים המציגים נתונים סטטיסטיים על השימוש בתוכנות, ציר זמן ההיסטורי המציג ניצול תוכנות ומשווה סטטיסטיות שימוש מוצברת בתוכנות. עוד ב-LABSTATS, היא תצוגה של דוחות התחברות סיכום התחברות על ידי משתמש ספציפי. השוואת של סטטיסטיות התחברות לפי משתמשים,-tag משתמש או קבוצת משתמשים במהלך פרק זמן מסוים. כמו כן גם לפי מחשב ספציפי, tag מחשב או קבוצת מחשבים במהלך פרק זמן מסוים ומוזהה את המספר המksamלי של הפעולות במקביל בו זמנית (נספח 23.3.3). בנוסף מציג ציר זמן ההיסטורי המציג את המספר המוצע של כניסה למחשבים, ממוצע היסטוריות שימוש לפי שבוע ומציג את אחוזי הניצול בחלוקת לפי שבוע או יום (נספח 23.3.4). בנוסף LABSTATS כולל מפות של מעבדות (נספח 23.3.5), כדי ליצירת מפות של מעבדות מחשבים והציג של זמינות של כל מחשב במעבדות אלה בזמן אמיתי.

Deep Freeze .8.2

תוכנת Deep Freeze מציעה את פתרון להגנה על עמדות עבודה. התוכנה יוצרת תמונה קופאה של הקונפיגורציות ושל הגדירות הרצויות בעמדות העבודה השונות כך שבמקרה של שינוי בلتאי רצוי המחשב יחזיר למצב ה"קפוא" הקודם שנבחר. מעבר להגנות שהתוכנה מספקת היא מספקת פתרון לניהול רשות מחשבים- חלוקה לוגית וдинאמית של הרשות, התקנות וערכונים אוטומטיים, שליטה מרוחק ומעקב. ישנו מעקב אחר שימושי המחשב, ניתן לקבל מידע אודוטה התוכנות בהם משתמשים במחשבים השונים ולהפיק דוחות לפי פרמטרים שונים, כגון מספר המשתמשים במעבדה (נספח 23.4.1), בעזרתו אוסף ומציגה התוכנה ניתן לראות את מצב השימוש של

המחשבים כדוגן עומס, תוכנות הנמצאות בשימוש רב, תוכנות שנרכש אליהם רישיון ולא נמצאות בשימוש וכן לא מצדיקות את הרכישה (נספח 23.4.2).

התוכנה יכולה להתאים לארגונים בעלי אלפי מחשבים, מעבדות, מחשבים רפואיים ובאותה מידת גם למחשב ביתי ותומכת במערכת הפעלה של MAC ו-Windows.

Kiwi Application monitor . 8.3

תוכנה לHECK אחר שימוש בתוכנות ופעולות משתמש. לאחר התקנת התוכנה במחשב יתחיל איסוף מידע אודות התוכנות והיישומים אותם מגדרים לה. ניתן לראות עבור כל יישום את זמן הריצה הממוצע ומקסימלי לשון, שימוש זיכרון וCPU ועוד (נספח 23.5.1). ניתן להגדיר שהתוכנה תרוץ ברקע עם הפעלת המחשב או בזמן מוגדר אחר .

ה kiwi בעיל יכולת התראעה ונקיית צעדים מוגדרים כתגובה לאירוע בזמן אמת, בכך מתאפשרים תהליכי אוטומטיים רבים לא התערבות המשתמש. ניתן לתת לתוכנה לנוהל את התהליכים הרצים, השירותים המותקנים, כאשר כל פעולה הנעשית על ידי kiwi תישמר בקובץ הלוג, או רק לאוסף נתונים נתונים אלו-חלק מהפיצרים אותם מציעה התוכנה הם : זמן ריצת יישום, שימוש זיכרון של היישומים השונים (נספח 23.5.2), תזוזת העכבר ומקלדת, השוואת יישומים, ניתן להגדיר את התאריכים עברים נרצה להציג את המידע וצורת הציגה (לדוגמא תרשימים פאי או גרף עמודות).

בנוסף קיימת תוכנה משלימה kiwi admin, החולשת על מספר Kiwi Application monitor (KAM) בראשת (נספח 23.5.3), ל-admin זה גישה להגדרות, ולנתוני שימוש של המחשבים תחת פיקוחו. באפשרותו לראות עדכונים, שגיאות והתראות(נספח 4) ואיסוף המידע למטרות סטטיסטיות.

8.4. סיכום מערכות קיימות

לאחר שסקרנו את המערכות הדומות והחמות בשוק נשווה ביניהן ובין המערכת המוצעת -
: CAMS

טבלה 1 - מערכות קיימות

CAMS	LABSTATS	Deep Freeze	KAM	דרישה
9	3	3	3	עלות
9	9	3	3	תצוגת מפות כיתה
9	9	6	6	הוצאת דוחות סטטיסטיים וגרפים
9	3	3	9	תמיכה במערכות הפעלה Windows ו-Linux
9	9	9	3	גישה web-ית
45	33	24	24	סה"כ

מקרה :

התאמה גבוהה - ניקוד 9.



התאמה בינונית - ניקוד 6.



התאמה נמוכה - ניקוד 3.



8.4.1. מסקנות

מסקירת הממערכות הקיימות והשוואות ראיינו כי הן לא מספקות את הדרישות הבסיסיים של המערכת הנדרשת, בנוסף הממערכות הקיימות גדולות ומכילות פונקציונליות רבה שלא רלוונטיות לארגון שלנו דבר יגרום עומס וסיבוכיות לא הכרחית. מסקנתנו היא שהפתרון הטוב ביותר עבור הל��ות הוא פיתוח מערכת CAMS שתעמוד בכל דרישותינו וצריכיו. במקומות להתאים את הלകות למערכת נוכל להתאים את המערכת ללകות.

9. טכנולוגיות

9.1. טכנולוגיות כלל

כדי לבנות את המערכת בצורה המיטבית והנכונה ביותר, סקרנו את הטכנולוגיות ושפota הפיתוח האפשריות למימוש הפרויקט, נציג את הטכנולוגיות אותן שקלנו לאחר סינון ראשוני, את היתרונות והחסרונות של כל אחת מהן ולבסוף את הטכנולוגיה הנבחרת שמספקת את הדרישות שלנו באופן אופטימלי.

קודם כל נציג את הדרישות שלנו :

- א. טכנולוגיה המאפשרת הפרדה בין משק המשמש לוגיקה לצורכי מודולריות המערכת, כיוון שהעובדת היא מול מסד נתונים שיש לאחסן בו נתונים רבים ושלוח מידע לעיתמים תcornות יש צורך שהטכנולוגיה תתמוך בעבודה נוחה מול בסיס נתונים.
- ב. על המערכת לעבוד עם מערכת ההפעלה של המחשבים באוניברסיטה, כדי לתקשר עם המחשבים ולאסוף מידע אנו זוקקות לטכנולוגיה או API שיאפשר לנו התממשקות נוחה וגישה למידע במערכות windows ו-Linux.
- ג. על המערכת לספק משק ניהול. משק זה יכול ניהול המחשבים לשם כך נציג טכנולוגיה שתאפשר משק ניהול אינטראקטיבי ונוח.
- ד. המשק הניל צריך לכלול הצגת סטטיסטיות וגרפים מותאמים אישית.
- ה. יש צורך בטכנולוגיה שתאפשר לנו יכולת תמרון נוחה וგמישה עם המידע מבסיס הנתונים עם כלים לעיבוד מידע מבסיס הנתונים ויכולת הצגתו בתרשימים וגרפים.
- ו. על המערכת להתממשק עם מערכת שיבוץ חדרים באוניברסיטה.
- ז. על המערכת (על פי דרישת הלקוח) לאפשר גישה web-ית לנין צריך שפה שתתמוך web.

- .ח. על המערכת **לפעול עם מערכות הפעלה שונות כגון Linux ו Microsoft Windows**
- .ט. שפת התכנות שנבחר צריכה להיות מוכרת ונתמכת ולא שפה שהיא "אקסקלוסיבית" על מנת שתהייה ניתנת לתחזוקה ע"י אנשים אחרים וניתן יהיה להתאים אותה במידת הצורך לצרכים עתידיים.

ASP.NET MVC .9.2

טכנולוגיית קוד פתוח ותיקה ומתקדמת לפיתוח יישומי רשות מבית מיקרוסופט. באמצעו ASP.NET ניתן ליצור אתר אינטרנט דינמיים, מבוססי אירועים, אשר תומכים בטכנולוגיות متقدמות כגון AJAX ועובדת מול בסיסי נתונים. הפלטפורמה מותאמת לארכיטקטורת MVC המפרידה בין שכבות האפליקציה ומאפשרת תכונות מודולרי. ASP.NET מבוססת על framework.NET של מיקרוסופט וכוללת מחלקות רבות לטיפול נתונים כגון מסדי נתונים, קבצי טקסט וקבצים גרפיים. ניתן לכתוב דפי ASP.NET בכל שפה בסביבת.NET. כמו C# או VB.NET.

יתרונות :

- .א. עבודה נוחה מול בסיס נתונים.
- .ב. התממשקות נוחה עם טכנולוגיות אחרות מבית מיקרוסופט כמו active directory ו-SQL server.
- .ג. כלים לתכנות ויזואלי.
- .ד. אפשרות לתכנות מודולרי באמצעות MVC.
- .ה. סביבת עבודה בעלת תמיכה טובה לביצוע טסטים ודיבוג.
- .ו. כיוון שעבדנו בעבר עם טכנולוגיה זו והיא מוכרת לנו לא נדרש למידת שפה חדשה.

חסרונות :

- .א. הריצה מוגבלת לשרת IIS (שרת האינטרנט של מיקרוסופט).
- .ב. יכולות גרפיות מוגבלות יחסית.
- .ג. דורשת רישיון על מנת לפתח בה, לא חינמית.

Django .9.3

טכנולוגיית קוד פתוח חינמית, זהה לENVironment שבסיסת python המעודדת פיתוח מהיר ויעקוב נקי לישומי web. מטרתה לתת למפתח להתמקד בפיתוח האפליקציה על ידי טיפול בסוגיות שונות אחורי הקלעים (כמו authentication של משתמשים ובדיקה וטיפול מובנה של פרצות אבטחה).

Django יש object-relational mapper (ORM) המאפשר למפות את מסד הנתונים לאובייקטים של python. ישנה חלוקה לשכבות- model, template ו-view. זהה השכבה הלוגית, view מגדר איך מידע מוצג וה-template אחראי על אופן תצוגת מידע זה, חלוקה זו מזכירה את תבנית MVC אך יש חלוקה ברורה בין הנתונים המוצגים ואופן הצגתם מה שמאפשר יכולת תמרון גדולה יותר בכל הנוגע לתצוגה. הסביבה עצמה מספקת

את ה"בקר" בעזרת קונפיגורציה מתאימה- קלומר חוסכת מההמפתח כתיבת שכבה זו וטיפול בה.

יתרונות :

- .א. עבודה נוחה מול בסיס נתונים.
- .ב. כלים מובנים ואוטומטיים לשימירה על אבטחה.
- .ג. כלים אוטומטיים להורדת הסביבות בפיתוח אפליקציות web.
- .ד. חינמית.
- .ה. קלה ו互動ית ללמידה.

חסרונות :

- .א. מונוליטי- הטכנולוגיה מכילה בתוכה קוד בסיס רחב המטפל במגוון מקרים, אומנם דבר זה חוסך הרבה עבודה מההמפתח אך זה כרוך בהנחות לגבי האינטראקציה של המרכיבים בקוד, לעומת זאת סביבה מודולרית המתחלקת להרבה מרכיבים בעלי מטרת מוגדרת, סביבה מונוליטית היא "גוש" אחד מה שליעיתים יוצר בעיות.
- .ב. קושי בייצירת טסטים וдинמיות.
- .ג. מבוסס Web Server Gateway Interface (WSGI)- פרוטוקול סינכרוני בעל מגבלה לטפל בבקשת אחת בלבד בכל זמן נתון.

Ruby on Rails (RoR) . 9.4

זהו סביבת פיתוח המותאמת לפיתוח אפליקציות רשת, מבוססת על שפת התכנות רובי- MVC. RoR מיישמת את ארכיטקטורת MVC מונחית-עצמים טהורה. MySQL, PostgreSQL, SQLite, היא תומכת ב- ORM. מיפוי אובייקטים למסד נתונים, RoR מאפשרת כתיבת אפליקציות רשת בעזרת מעט מאוד שורות קוד בעזרת פשטותה ועקרונותיה :

Don't Repeat Yourself -DRY, מניעת חזרה על קוד. Convention over Configuration -CoC, יש להגדיר רק דברים יוצאי דופן.

לRoR תמיכה במנוע שירותים SOAP ממקור האפליקציה, ספריות JavaScript לעובדה קלה עם AJAX ועוד.

יתרונות :

- .א. תכונות בסטנדרטים גבוהים- שימירה על מודולריות באמצעות ארכיטקטורת MVC, עקרון DRY.
- .ב. עבודה נוחה מול בסיס נתונים.
- .ג. תמיכת טובה בtesteים אוטומטיים.
- .ד. סביבה מובנת לביצוע טסטים.
- .ה. חינמי.

חסרונות:

- א. זמני ריצה וביצועים- זמן העיבוד הוא איטי לעומת שפות אחרות ולא מותאים לאתרים שצרכיהם הרבה חישובים במקביל, הרבה פעולות מורכבות נמצאות ב"מאחורי הקלעים" של פונקציות פשוטות.
- ב. על מנת לבצע דיבוג יש להוריד ולהתקין ספריות נוספות, זהו לא כלי מובנה או אינטואיטיבי.
- ג. דוקומנטציה- אין דוקומנטציה מספקת טוביה של הספריות והכלים, במיוחד אם מדובר בספריות ייחודיות.
- ד. זמן לימוד השפה- המוסכמויות בשפת רוביו הן שונות ממה שאנו מכירים מתכנות בשפות אחרות ולכן יש צורך לתקופת במידה ארוכה לחסית עד לשיליטה בה.

Web2py . 9.5

סבירות עבודה חינמית לפיתוח מבוססת python, היא מותאמת למתחדשות הטכנולוגית agile, ומתחילה לפיתוח full stack של אפליקציות web. מיישמת את ארכיטקטורת MVC כך שיש חלוקה ברורה בין שכבה הלוגית, שכבת התצוגה ושכבת הבקרה. ישנו דגש על אבטחה- ישנו טיפול אוטומטי בנקודות תורפה שיכולות לגרום לפרצת אבטחה (למשל בדיקת קלט על מנת למנוע SQL injection).

סבירות עבודה זו נוחה לעבודה מול בסיס נתונים ומתחילה לאפליקציות מבוססות מסדי נתונים—(database-driven) ישנה שכבת אבטר��ציה שמתקשרת עם בסיס הנתונים באופן דינامي, תומכת בatabase ,PostgreSQL ,MySQL ,Oracle ,SQLite ועוד.

 יתרונות :

- א. עבודה נוחה מול בסיס נתונים.
- ב. שימוש בـMVC.
- ג. כלים מובנים ואוטומטיים לשימירה על אבטחה.
- ד. קלה ואינטואיטיבית ללמידה.
- ה. אופטימיזציה של הקוד להורדת זמן ריצה ושימושי זיכרונו.

חסרונות:

- א. אין תמיכה מספקת טוביה בטסטים.
- ב. דיווח שגיאות לא נוח- מבחינת התצוגה ו מבחינת המידע הנitinן למפתח אוזות השגיאה, דבר המקשה על תהליך הפיתוח.
- ג. אומנם מתחילה עבודה נוחה עם בסיס נתונים אבל דרך הגדרת השדות והרשומות היא פונקציונלית ולא object oriented מה שיכול להתאים למודל קטן יחסית אך במודל גדול זה הופך לקשה לתחזוקה.
- ד. לא ניתן להשתמש בכלים פיתוח סטנדרטיים של python ללא התאמה מיוחדת.

 9.6. סיכום טכנולוגיות

נסכם את הטכנולוגיות אותן סקרו ונערוך השוואה :

טבלה 2 – טכנולוגיות

דרישה	ASP.NET MVC	Django	Ruby on Rails	Web2py
תמייה בweb	9	9	9	9
כלים לביצוע בדיקות ובדיקה	3	6	6	3
עובדת מול בסיס נתונים	9	9	9	9
מודולריות	9	9	9	9
הتمמשקות עם מערכות windows	3	3	3	9
תצוגה גרפית	9	9	9	6
עלות	9	9	9	3
היכרות עם הטכנולוגיה והקצתה משאים למידה	6	3	6	9
סה"כ	63	60	57	57

מקרה :

התאמה גבוהה - ניקוד 9.



התאמה בינונית - ניקוד 6.



התאמה נמוכה - ניקוד 3.



9.6.1. מסקנות

לאחר השוואת הגענו למסקנה כי ASP.NET MVC היא הפלטפורמה הטובה ביותר עבורנו, היא עומדת בכל הקריטריונים למעט תצוגה גרפית מתקדמת, על קריטריון זה נוכל להתגבר באמצעות כלי תצוגה לצד הלקוח. הפיתוח הוא לא חינמי אך כיוון שיש לנו רישיון חינמי כסטודנטיות, קריטריון זה לא מהויה מוגבלת. בגלל שמדובר בטכנולוגיה מבית מיקרוסופט היא אידיאלית עם עבודה והتمמשקות עם כלים אחרים מבית מיקרוסופט. פרט אינטגרלי בפרויקט שלנו כיוון שאת רוב המידע אנו מחלcout ממערכת הפעלה של המחשבים באוניברסיטה. בנוסף יש לנו ניסיון בתכנות ב-C# דבר שנตอน יתרון על יתר הטכנולוגיות ששקלנו והוא תיתן המשכיות לפרויקט שלנו ותחזקתו לאחר ההטמעה דבר שלא בטוח היה מתקיים אם היו בוחרות בטכנולוגיה יותר "אקסלוסיבית" וייחודית.

10. בחירת בסיס נתונים

כעת כשבחרנו את סביבת הפיתוח והשפה בה נעבד, علينا להחליט על בסיס נתונים. נרצה בסיס נתונים רלוונטי כיון שארגון הנתונים באופן טבלאי הוא הנוח ביותר ומתאים לנו.

Microsoft SQL Server .10.1

תוכנה מבית מיקרוסופט לניהולiesel מאובטח של בסיס נתונים בצורה רלוונטית. התקשרות עם בסיס הנתונים המנוהל MSSQL מותאפשרת בעזרת שאילתות -פקודות בשפת SQL, שפה עשירה בעלת פקודות רבות המאפשרת כתיבת שאילתות פשוטות ומורכבות המשמשות ליצירת טבלאות והיחסים ביניהן, לשאול וטיפול נתונים. השפה בה מועברות הפקודות למערכת היא Transact-SQL, מימוש של SQL בתוספת הרחבות לשפה הסטנדרטית.

יתרונות :

- .א. רמת ביצועים גבוהה.
- .ב. סטנדרט אבטחה גבוהה.
- .ג. עובד טוב עם מוצריו של Microsoft - גישה נוחה לבסיס נתונים מ-.NET-ASP.
- .ד. שירותים תומכי טובים.

חסרונות :

- .א. עלות גבוהה של שרת נתונים
- .ב. דרישות חומרה גבוהה.

Oracle .10.2

מערכת לניהול בסיס נתונים בצורה רלוונטית שפותחה על ידי אורקל. ישנה אפשרות לכתוב לאחסן ולהריץ פרוצדורות ופונקציות בתוך בסיס הנתונים. בסיס הנתונים תומך ביום במספר שפות PL/SQL – זהה שפה פרוצדורלית קניינית של חברת אורקל או בשימוש בשפת תכנות מונחית עצמים Java - אשר מופעלת שירותי מתוך מסד הנתונים. ישנה אפשרות נוספת של כתיבת קוד חיצוני לבסיס נתונים באמצעות השפה C או פסקל ולהשתמש בהן מתוך בסיס הנתונים. ישנה שמירה על אוטומיות - כל טרנסקציה שמתבצעת מתבצעת בשלמותה כך שיש שמירה מלאה על שלמות ותקינות הנתונים.

יתרונות :

- .א. אוטומיות.
- .ב.יעילות ורמת ביצועים גבוהה.
- .ג. סטנדרט אבטחה גבוהה.
- .ד. שירותים תומכי טובים.

חסרונות :

- .א. עלות גבוהה של שרת נתונים.

- ב. דרישות חומרה גבוהות.
ג. ידע מוקדם ב-PL/SQL.

10.3 PostgreSQL

מערכת קוד פתוח לניהול בסיס נתונים רלציוני חופשית שאינה נשלטת על ידי אף חברת אלה נתמכת על ידי קהילה גlobלית של מתכנתים וחברות שפיתחו אותה. רצה על כל מערכות הפעלה העיקריות (windows, Linux, MAC). המערכת משתמשת Multi-Version Concurrency Control (MVCC) (MVCC), כלומר העבודה על בסיס הנתונים מתבצעת בגרסאות לכך שניתן לעקב אחר שינויים, לראות את מצב הנתונים בנקודת זמן ספציפית ולבצע שחזורים במידת הצורך. המערכת היא מאוד 'סקלבילית' גם מבחינת כמות הנתונים אותן היא יכולה לתמוך וגם מבחינת כמות המשתמשים הניגשים לבסיס הנתונים.

יתרונות :

- א. סקלביבלי.
- ב. יכולה לרוץ על פלטפורמות רבות.
- ג. חינמי.

חסרונות :

- א. דוקומנטציה לא טובہ.
- ב. המהירות נפגעת בפעולות בנפח גדול.
- ג. קושי בכתיבה קונפיגורציה.

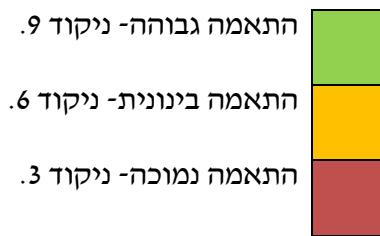
10.4 סיכום בסיסי נתונים

נארוך השוואה בין בסיסי הנתונים השונים :

טבלה 3 - בסיסי נתונים

דרישה	PostgreSQL	Oracle	MS SQL Server
שמירה בצורה טבלאית	9	9	9
עובדת נוחה מול .NET.	6	6	9
יעילות	3	9	9
דרישות חומרה	9	3	3
דוקומנטציה ושירותי תמיכה	3	9	9
עלות	9	3	3
היכרות עם הטכנולוגיה והקצת משאבי למידה	6	3	9
סה"כ	45	42	51

מקרה :

**10.4.1. מסקנות**

על המערכת שלנו להיות בקשר תמידי עם בסיס הנתונים, לאחסן הנתונים, לשיליפת וניתוחם لكن נרצה בסיס נתונים יעיל ככל האפשר שישמר על שלמות ואמינות המידע. בנוסף חשוב לנו שייהיה נוח לעבוד עם בסיס הנתונים מסביבה העובדה בה נפתח את המוצר- ASP.NET-MVC- בסיס הנתונים שעומד בדרישה זו הכי טוב הוא MSSQL יתרון נוסף הוא העובדה שיש לנו היכולות עם בסיס נתונים זה. החלנו שהשימוש בבסיס הנתונים של מיקרוסופט הוא הבחירה הטובה ביותר עבורנו- למרות העלות גבוהה רלוונטיות כלל בגל היותר סטודנטיות) ודרישות החומרה.

11. בחירת מתודולוגיית פיתוח

מתודולוגיית פיתוח תוכנה היא אוסף מוסכם של עקרונות וכליים על פי הם מפותחות ומתחזקות מערכות תוכנה. בתחום התוכנה קיימות מתודולוגיות שונות ומגוונות המתיחסות לדריכים שונים לעקרונות המנחים, יישום פרויקט תוכנה ואופי ניהולו, על מנת לבנות מוצר טוב צריך שהמתודולוגיה תתאים לצרכי הפרויקט והלകות.

11.1. מודל מפל המים

זהו מודולוגיה השיכת למשפחת המתודולוגיות הקומיות- משפחה זו מבקשת ליחס את איקות התוצר ההנדסי לתהליכי הפיתוח של התוכנה כולם כדי שתוצר יעמוד בקריטריונים של אמינות, עמידות ודיוק על תהליך הפיתוח לעמוד בקריטריונים אלה. מודל של פיתוח תוכנה מסורתי משנו ה-70 המתבסס על העיקרון שככל שלב בפיתוח מתבצע לאחר שקודמו הסתיים.

שלבים מרכזיים- איסוף וניתוח הדרישות, עיצוב תוכנה, בדיקות, שילוב, הטמעה ותחזקה.

תהליך העבודה- טורי, לא מתחלים שלב לפני שטبيعيים את השלב הקודם לו, כיון שככל שלב מושפע מהשלבים הקודמים. בכל שלב יש מיקוד במשימה עיקרית אחת בלבד. ניתן לחזור אחרה במידה ויש צורך בתיקונים.

דגש- במתודולוגיה זו דגש רב על איסוף הנתונים, אפיון, תכנון ועיצוב- לפי גישה זו ניתן לתוכנן ולנהל פרויקט פיתוח באופן סדר.

עלותTeVויות וتكلות- אם מתגללה תקללה בשלב מאוחר עלותה יכולה להיות, אך אם היא מתגללה בשלבים הראשונים ככל שהשלב מוקדם יותר, עלות התקיקון תהיה נמוכה יותר. הדגש על הניתנות ועיצוב המערכת בשלבים מוקדים אמר לעוזר לאתר תקלות ולמנוע אותן.

קשר עם הלকות- אופי ניהול הפרויקט יוצר נתק בין המפתחים ללקוח או משתמשי הקצה- זהו

תהליך ארוך ואין שמירה על קשר רציף עם משתמשי התוכנה.
מתודולוגיה זו מותאמת לפרויקטים גדולים אשר זמן הפיתוח שלהם הוא ארוך.

Scrum . 11.2

מתודולוגיה השיכת למשפחת המתודולוגיות הזריזות (Agile) - אלו הן מתודולוגיות איטראטיביות שמתאימות לפיתוח בצוותים קטנים במתן דגש על יעילות, זריזות, אינטואיטיביות ושיתופי פעולה מכובן מטרה.
מתודולוגיית Scrum יוצאת מנקודת הנחה שלא ניתן לחזות, לתכנן או להגדיר כל פרט לפני שלב הפיתוח.

שלבים מרכזיים - כל איטראציה מכילה אפיון, עיצוב, בניה ובדיקות.

תהליך העבודה - תהליך הפיתוח מתבצע באיטראציות של כחדר שכך כל איטראציה היא תהליכי הפיתוח שמתחל שוב מראשיתו, פעם בחודש מספקת תוכנה עובדת למשתמשים, והערכותיהם, כמו גם הדרישות החדשנות, מישומנות במחזור הפיתוח הבאים. בנוסף ישנים ספרינטים יומיים - בכל עשרים וארבע שעות מתבצעות מטלות פנימיות של הצוות שנסקרות בפגישה היומית.

דגש - המודל מתאים לצוותים קטנים, שם דגש על יעילות, זריזות וaicות ומתמקד בשיפור היכולת להגיב לשינוי בדרישות תוך כדי הפיתוח ואספקת מוצר בהירותה בשלב מוקדם ללקוח. לכל ספרינט יש פגש תכנון - הגדרת פרטי העתודה לספרינט ופגש ניתוח - הפקת ליקחים.

עלות טיעות ותקלות - גם במתודולוגיה זו ישנה עדיפות לאייתור בעיות מוקדם אך יש לה עמידות גבוהה יותר ויכולת התמודדות טוביה יותר עם תקלות.

קשר עם הלקוח - בשיטה זו יש קשר יומיומי בין המפתחים ללקוח, והתוצר הוא עבודה צוות משותפת ביניהם.

מודול זה מותאים לפרויקטים קטנים יחסית שימוש זמן פיתוחם הוא קצר יותר.

Unified Process . 11.3

מתודולוגיה השיכת למשפחת המתודולוגיות האיטראטיביות, זהה משפחה של מתודולוגיות שרואות בפיתוח תוכנה תהליכי מחזורי או ספירלי בו התוכנה נבנית בהדרגה. במשפחה זו דגש על כלים, ארכיטקטורה, ניהול תצורה ופיתוח החלקים המורכבים מוקדם ככל האפשר על מנת להפחית את הסיכון בפיתוח.

מתודולוגיית Unified process (UP) אינה מתודולוגיה בסגנון הקלאסי אלא מסגרת מתודולוגית הנינתנת להרחבה ויש להתאים לפרויקט, היא מורה על פיתוח תוכנה בתהליכי מחזורי רבים שלבים מרכזיים - חלוקה לארבעה שלבים עיקריים :

התחלת-בסיס המודל העסקי, הגדרת תרחישי שימוש, דרישות, זיהוי סיכון והכנות לוח זמנים. שלול- שלב זה מכסה את רוב דרישות המערכת, עיקר השלב הוא עיצוב המערכת, פיתוח בסיסי הכלול את פונקציונליות המפתח של המערכת ויצירת תכנית לשלב הבניה.

בנייה- בשלב זה מפותחים כל חלקים המערכת שלא נכנסו לשכלול, נהוג לחלק את העבודה לתרחישי שימוש כך שכל תרחיש שימוש הוא התחלת של איטראציה חדשה.

מעבר- בשלב זה המערכת מועברת לקבוצת משתמשים מצומצמת ומפותח אימון המשתמשים על המערכת, מפותחים תיקונים ושיפורים על סמך המשתמשים המתקבלים מהמשתמשים.

תהליכי העבודה- כל שלב מוחלך לאיטראציות, רוב האיטראציות יכולו לעבודה על כל תחומי התהילה (דרישות, עיצוב,นำไปש ובדיקות) אך המאמץ והדגש היחסי בין התחומים השתנה במהלך התקדמות הפרויקט. החל משלב הבניה על כל איטראציה להסתים בגרסה עובדת. הטכניקות העיקריות של המתודולוגיה כוללות פיתוח מחזורי תחום-זמן, ניהול השינויים באמצעות כלי ניהול שינוי ובקורת תצורה, ניהולדרישות ומידול ויזואלי באמצעות UML. **דגש**- ניהול הפרויקט, ניהול שינויים, ניהול סיכון, ארכיטקטורה ותהליכי המידול של התוכנה.

עלותTeVיות ותקלות- כיוון שהמתודולוגיה ממוקדת סיכון, התוצרים בעלי הסיכון הגבוה מפותחים בשלב מוקדם של מחזור חי הפרויקט. אומנם למתודולוגיה יכולת התמודדות טובה עם שינויים ותקלות, השאיפה היא לגנות בעיות כמה שיותר מוקדם.

קשר עם הלקוח- נוצר נתק בין המפתחים ללקוח או משתמשי הקצה עד לשלב מאוחר יחסית בפיתוח- שלב המעבר שرك בו ניתן המוצר למספר משתמשים ומתබל משוב בהתאם אליו נמשך תהליך הפיתוח.

מודל זה מתאים לסוגים גדולים רבים של פרויקטים לפיתוח תוכנה, החל מפרויקטים קטנים ולא קריטיים ועד לפרויקטים גדולים לפיתוח מערכות קריטיות תומכות חיים.

11.4. סיכום מתודולוגיות פיתוח

טבלה 4 - מתודולוגיות פיתוח

תבונה	מפל המים	SCRUM	Unified process
תגובה לשינויים	3	9	6
דגש על תיעוד	9	3	6
דגש על קוד עובד	3	9	9
דגש על בדיקות אוטומטיות	3	9	6
דגש על ניהול סיכון	3	3	9
התאמה לגודל הפרויקט	3	9	9
פשטות הטעינה	9	3	6
התאמה משך איטראציה	3	6	9
סה"כ	36	51	60

מקרה :

התאמה גבוהה- ניקוד 9.

התאמה בינונית- ניקוד 6.

התאמה נמוכה- ניקוד 3.

11.4.1. מסקנות:

בבחינת מתודולוגיות ננויל מול דרישות הפרויקט ורמת מורכבותו הגיעו למסקנה שהמתודולוגיה שהכחית מתאימה לנו היא UP מהסיבות הבאות:

- א. מתודולוגיה זו שמה דגש על הבנת הצרכים והדרישות בתחילת הדרך כך שהפיתוח מתחילה כאשר המטרות ברורות ומוגדרות היטב, אך אופן הפיתוח מאפשר עדכונים בהמשך הדרך בעקבות שינויים. לעתנו זהו שילוב אופטימלי של ניתוח מקדים וIMPLEMENTATION לשינויים לפרויקט בסדר גודל שלנו.
- ב. תהליך הפיתוח מתחילה בפיתוח החלקים המרכזיים והקריטיים ביוטר למערכת- שלב השכלול, מה שמתאים מאוד לפרויקט שלנו כיון שכל המערכת שלנו מtabסת על איסוף הנתונים- וזהו התהליך המרכזי ביוטר שלנו ולאחר פיתוחו יתווסף שאר הfonקציונליות באיטראציות כחלק משלב הבניה.
- ג. מתודולוגיה זו היא בעצם מסגרת מתודולוגית אותה ניתן להဏים לצרכי הפרויקט, למשל את משך ואופי האיטראציות ואת רמת הקשר עם הלוקה.
- ד. דגש חשוב במתודולוגיה זו הוא מיקוד סיכון- נשים לב לסיכון קריטיים כמה יותר מוקדם בחנייה מוחזק הפרויקט וסדר הפיתוח יקבע כך שהותוצרים בעלי הסיכון הגבוה יהיו ראשונים כך נבטיח טיפול בחלקים הבעייתיים ביוטר בהתחלה.

12. סיכום חלק ראשון

בפרק זה הצגנו את המרכיבות הקיימות ביום אוניברסיטה ל>Showcases תפוסות כיתות מחשוב ונתונים סטטיסטיים עבור נציגות תפוסת כיתות מחשוב, הראינו את החולשות במצב הקיימים והצריך במערכת חדשה. בחנו מערכות דומות שקיימות בשוק, השווינו ביניהן בין הצרכים של הלוקה והגענו למסקנה שיש צורך בפיתוח מערכת חדשה.

ניתחנו את הצרכים הטכנולוגיים של המערכת וסקרנו מספר טכנולוגיות קיימות בשוק, כמו כן בסיס נתונים ומתודולוגיות פיתוח. לאחר בדיקה והשוואה החלנו שנבעוד בטכנולוגיית Microsoft SQL עם בסיס נתונים ASP.NET MVC .Unified process

13. ניתוח ועיצוב המערכת

לאחר שניתחנו את המערכת הקיימת, בדקנו מה החולשות שלה, הגדרנו את הצרכים של המערכת החדשה ובחרנו את הטכנולוגיות לפיתוח מתוך הטכנולוגיות שסקרנו, נטמך בניתוח ובעיצוב של המערכת המשודרגת.

בפרק זה נציג את דרישות המערכת, את הfonקציות אותן היא תכלול ואופן מימושן, את החלקים שמרכזים את המערכת ואת האופן שבו הם מתקשרים בינהם.

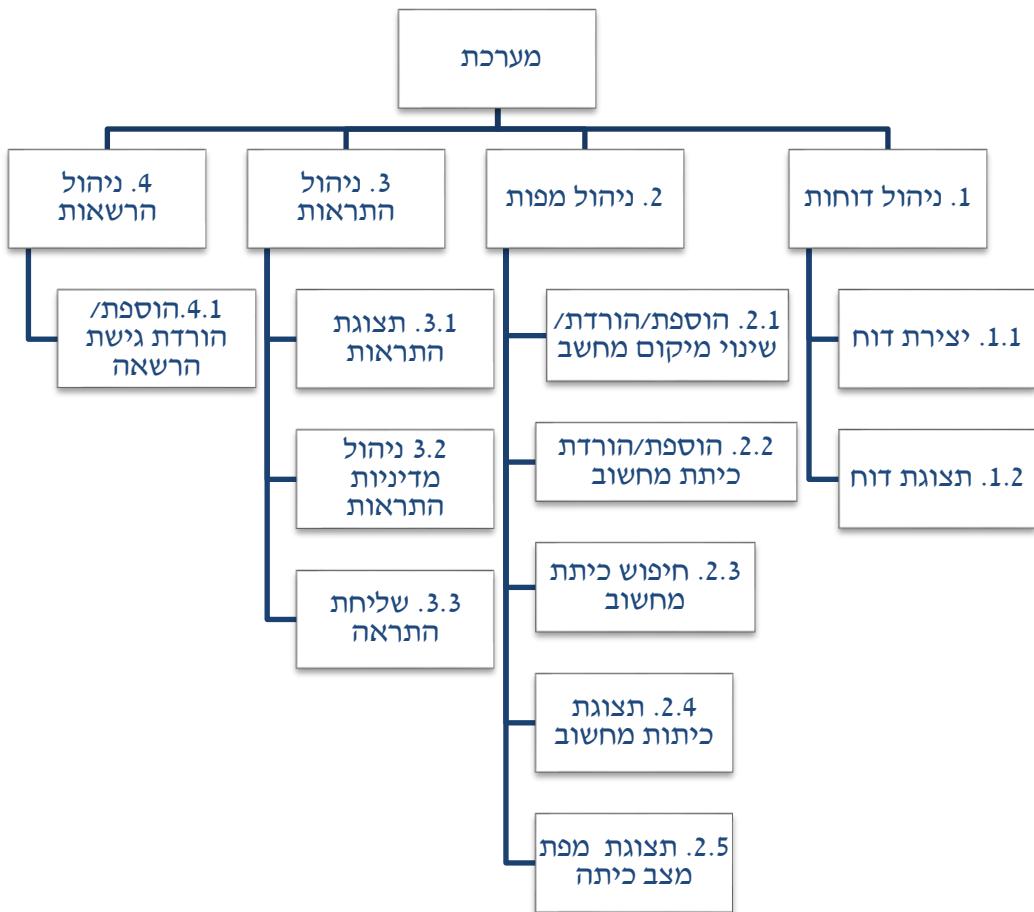
מטרתנו היא לפתח מערכת נוחה, יעילה ושימושית למעקב אחר פעילות המחשבים ברחבי האוניברסיטה ולשם כך נרצה שהתהליכיים יהיו פשוטים ככל האפשר וgemäßים לשינויים.

לצורך הבנת מבנה המערכת העתידית יצרנו תרחישי שימוש, תרשימי זרימות מידע (DFD) ותרשים מחלקות.

14. אפיון ראשוני של המערכת

14.1. דרישות

14.1.1. עץ דרישות



איור 1 - עץ דרישות

14.1.2. דרישות עיקריות

א. ניהול דוחות:

- משתמש יוכל לבחור עבור איזה כיתת מחשב או קבוצת כיתות מחשב ירצה להפיק דוח.
- רק משתמש בעל הרשות גישה לכיתת מחשב יוכל להפיק דוח אודוט כיתה זו או אודוט מחשבים בכיתה.
- בעת יצירת הדוח למשתמש תהיה אפשרות להגדיר כי אם בכיתה מחשבים משובץ שיעור יש להחשב את כל המחשבים בכיתה כתופסים.

א.4. בעת יצירת הדוח למשתמש תהיה אפשרות להגדיר כי לא תהיה התחשבות בשיבוץ השיעורים אלא יוצגו נתוניים על השימוש בפועל.

א.5. ביצירת דוח לכיתת מחשב יילקחו בחשבון רק המחשבים שהיו בכיתת מחשוב זו בתאריכים המבוקשים.

א.6. בתצוגת הדוח המשמש יוכל לבחור לקבל דוח פירוט על כיתה ספציפית.

א.7. הדוח יציג בטבלה ובנוסף על גרפ שמציג את הנתוניים.

ב. ניהול מפות:

ב.1. לכל מחשב במטה יהיה איקון בצורת מחשב עם שם המחשב שמייצג אותו.

ב.2. על מנת לשנות מקום מחשב במטה המשמש יגרור את איקון המחשב למיקום המתאים.

ב.3. המשמש יוכל להוסיף מחשב לכיתת מחשב מתוך הirectory active directory.

ב.4. לכל מחשב תהיה לכל יותר כיתת מחשב אחת אליה הוא שייך בכל זמן נתון.

ב.5. לכל מחשב תהיה היסטוריה מקומות, שתעדכן בכל הוספה/הסרה/שינוי מקום המחשב.

ב.6. רק משתמש בעל הרשות ניהול מלאה לכיתת מחשבים יוכל למחוק אותה.

ב.7. כל כיתת מחשב תהיה שייכת למחלקה אחת בלבד.

ב.8. רק משתמש בעל הרשות חלקית או מלאה למחלקה יוכל ליצור כיתת מחשב חדשה או לעורוך כיתת מחשב במחלקה זו.

ב.9. לכל כיתת מחשב יהיה מידע חד-חד ערכי, שם כיתה, מספר בניין ומספר כיתה.

ב.10. בעת כניסה למסך החיפוש מעבדות המחשב יוצגו על גבי מפת האוניברסיטה ע"י נקודות ציון.

ב.11. ניתן לחפש כיתות מחשב על ידי שורת החיפוש על ידי הכנסת טקסט חופשי, בחירת מחלקה, בחירת בניין.

ב.12. תצוגת מפת מצב כיתה תהיה זמינה ללא צורך בהתחברות.

ב.13. לכל מחשב במטה יהיה איקון בצורת מחשב עם שם המחשב שמייצג אותו.

ב.14. כל מחשב יהיה מסומן בצבע בהתאם למצב שלו. אדום- תפוס, ירוק- פנו, שחור- לא ידוע.

ב.15. מחשב תפוס הוא מחשב אשר יש משתמש המחבר אליו, או מחשב הנמצא בכיתת מחשוב בה משובץ שעורר בזמן הנוכחי.

ב.16. בכל מפת כיתה מחשוב יוצג לוח שעות השיעורים המתקיים באותו כיתה באותו יום.

ב.17. שעות השיעורים המתקיים בכיתות המחשב יילקחו מערכות שיבוץ החדרים של האוניברסיטה.

ב.18. מצב המחשבים יהיה עדכני עד שלוש דקות אחרת.

ג. ניהול התראות:

ג.1. בעמוד ההתראות יוצגו מחשבים אשר לא מגיבים או לא הבצעה אליהם התחברות תקופת זמן- אותה מגדר המשמש.

ג.2. עמוד ההתראות יתעדכן פעם ביום.

ג.3. לכל הטראה יצון שם המחשב, ניתנת המחשב בפה הוא נמצא, המחלקה אליו הוא שייך
ומספר הימים שמחשב זה לא הגיב/מספר הימים שלא הייתה התחברות למחשב זה.

ג.4. כל משתמש יוכל להגיד את מדיניות ההתראות הפרטית שלו.

ג.5. משתמש יגיד כמה ימים לאחר שמחשב לא מגיב יש להציג לו הטראה על כך.

ג.6. משתמש יגיד כמה ימים לאחר התבצעה התחברות למחשב יש להציג לו הטראה
על כך.

ג.7. משתמש יוכל להגיד על שליחת ההתראות למיל.

ג.8. התראות ישלו בהתאם להגדרת מדיניות המשתמשים- באופן יומי, שבועי, חודשי או
לא ויכללו את רשימת המחשבים אשר לא מגיבים או לא התבכעה אליהם התחברות
תקופת זמן- אותה מגדיר המשתמש.

ד. ניהול הרשות:

ד.1. משתמש בעל הרשות גישה צפיה יוכל לייצר דוחות על כוונות מחשב ומחלקות תחת
הרשותו.

ד.2. משתמש בעל הרשות גישה חלקית יוכל להוסיף, למחוק ולהזיז מחשבים בתוך כוונות
מחשוב תחת הרשותו.

ד.3. משתמש בעל הרשות גישה חלקית יוכל ליצור ולערוך כוונות מחשוב תחת הרשותו.

ד.4. משתמש בעל הרשות גישה מלאה יוכל למחוק כוונות מחשוב תחת הרשותו.

ד.5. הרשות גישה חלקית תכלול את כל הרשות של הרשות גישת צפיה.

ד.6. הרשות גישה מלאה תכלול את כל הרשות של הרשות גישה חלקית.

ד.7. גישת הרשות למחלקה תיתן גישת הרשות לכל אחת מכיוונות המחשב הנמצאות תחת
מחלקה זו.

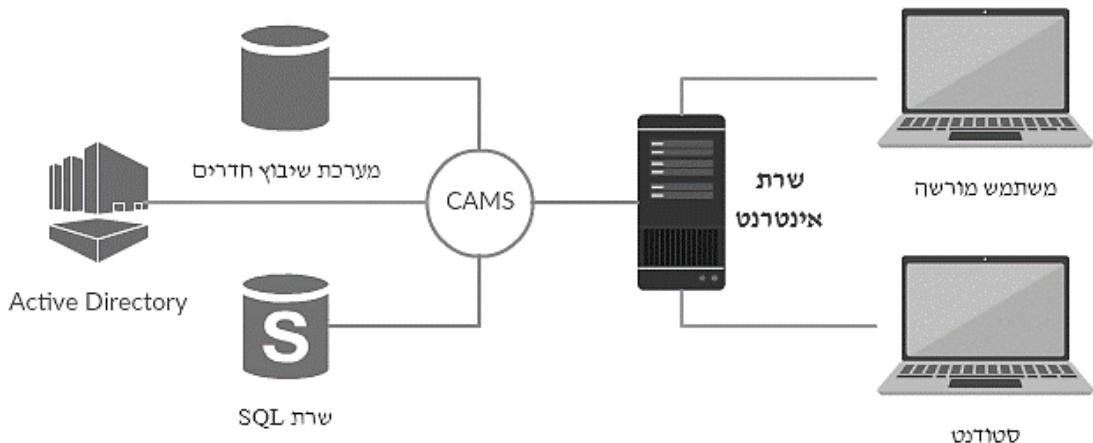
ד.8. רק משתמש בעל הרשות גישה מלאה למחלקה יוכל לתת הרשות גישה מכל סוג
למחלקה זו למשתמש אחר.

ד.9. משתמש בעל הרשות גישה מלאה למחלקה יוכל למחוק הרשות גישה של משתמש אחר
במחלקה זו.

15. ניתוח ועיצוב המערכת החדשה

15.1 ארכיטקטורה

אוניברסיטה בן גוריון



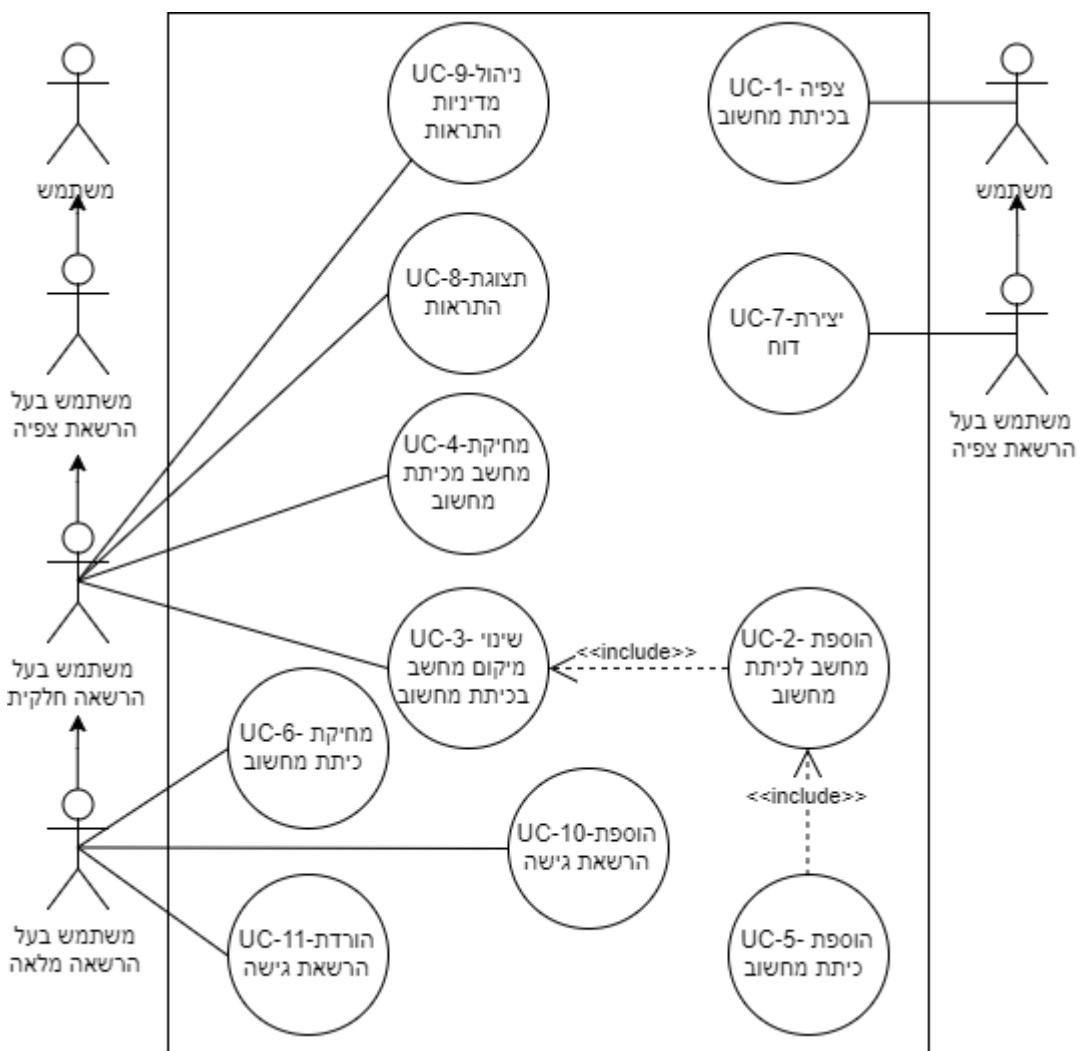
איור 2 - ארכיטקטורה

1. Active Directory – היא חבילה של כלי שירות שפותחה על ידי מיקרוסופט לניהול רשותה בארגונים. החבורה כוללת שירותי כגון: שירות ספרייה המבוססים על פרוטוקול LDAP, שירותי אימות המבוססים על פרוטוקול קרבווס. חבילת השירותים מאפשרת ניהול מרכזי של רשת המחשבים בארגונים. בעת הוספה מחשב לארגון מתבצע תהליך לצירוף לספרייה ניהול. חלק מהתהליך אתחול המחשב הוא מזדהה באמצעות מזהה ייחודי (GUID) מול הספרייה, המשמש במחשב מזדהה באמצעות שם משתמש וסיסמה (או כל שיטת אימות נוספת) ונקבעות עבורם כל ההרשאות הרלוונטיות.

כאשר משתמש מורה נכנס למערכת CAMS הוא מזין שם משתמש וסיסמה בארגוני בתאר אינטרנט. ההזדהות מהורי הקלעים מתרכחת מול AD, במידה ויש אישור המשתמש מועבר למערכת CAMS. בנוסף, כאשר מערכת CAMS אוספת נתונים על מצבם הנוכחי של המחשבים היא פונה לאד על מנת לקבל את הנתונים העדכניים הנדרשים על מצב המחשבים.

2. **שרת אינטרנט** – המשתמשים נגישים למערכת CAMS דרך שירותי שרת האינטרנט.
3. **CAMS** – מערכת ניטור פעילות מחשבים.
4. **שרת SQL** – מערכת CAMS קוראת וכותבת נתונים לבסיס הנתונים שנמצא על שרת SQL.
5. **משתמש מורה** – משתמש אשר מוגדר במערכת CAMS.
6. **סטודנט** – סטודנט או כל משתמש אחר שנכנס לכתובת האתר.
7. **מערכת שיבוץ חדרים** – מערכת הקיימת בארגון שממנה מערכת CAMS אוספת נתונים על שיבוץ חדרים באופן יומי.

.15.2 תרחישי שימוש



איך לתרגם מילים

15.12.1 UC-1 - צפיה בכיתת מחשב

תיאור כללי: משתמש יכול לצפות בכיתת מחשוב יבחר את הכיתה הרצויה ומפתח הכתובת תוצג לו.
শকনি: কল মশ্তমশ.

תנאי קדם: כיתת מחשוב קיימת.

תנאי סיום : מפת כויתת המחשבות תוצג על המסך.

תרמייש האלחה עיירוי:

- א. משתמש בוחר כייתת מחשוב בה מעוניין לצפות.

ב. שיליפת המחשבים השיעכניים לכיתת המחשוב הנבחרת.

ג. עברור כל מחשב בcytihah:

 - 1) הגדר את צבע אייקון המחשב בהתאם לפעולות הנווכית
 - 2) הגדר את מיקום אייקון המחשב בmphfa לפי שדה המיקום
 - 3) שיליפת השיעורים המתוכננים לכיתת מחשוב הנבחרת ביום הנהו

ד. עברור כל שיעור המותוכן לכיתת מחשוב זו:

 - 1) הוסיף את שעות השיעור למערכת היוםית של הכיתה.
 - 2) תציגות מפת כיתת המחשוב ומערכת יומית.

הרחבות : אין.

15.12.2 UC-2 - הוספה מחשב לכיתת מחשב

תיאור כללי: משתמש בעל הרשאה מתאימה יוכל לבחור מחשב להוספה לכיתת מחשב ומחשב זה ישוק לכיתה זו.

angkanim: משתמש בעל הרשאה חלקית/מלאה.

תנאי קדם: כיתת מחשב קיימת, המחשב הרצוי להוספה לא שייך לאף כיתת מחשב, למשתמש הרשות גישה מלאה/חלקית למחולקה אליה שייכת כיתת המחשב.

תנאי סיום: מחשב נוסף לכיתת מחשב.

תרחיש הצלחה עיקרי:

א. משתמש יבחר מחשב להוספה.

ב. המחשב יתווסף לרשימה המחשבים השبيחים לכיתת מחשב זו.

ג. ביצוע-3-UC-3- שיינוי מקום מחשב בכיתת מחשב.

הרחבות : אין.

15.12.3 UC-3 - שיינוי מקום מחשב בכיתת מחשב

תיאור כללי: משתמש בעל הרשאה מתאימה יוכל לשנות מקום מחשב בתוך כיתת מחשב באמצעות גיררת איקון המחשב במפה.

angkanim: 用户拥有对地图上某台计算机的管理权限。

תנאי קדם: כיתת מחשב קיימת, המחשב הרצוי להזזה שייך לכיתת מחשב זו, למשתמש הרשות גישה מלאה/חלקית למחולקה אליה שייכת כיתת המחשב.

תנאי סיום: מקום המחשב מתעדכן למיקום אליו נגרר.

תרחיש הצלחה עיקרי:

א. משתמש לוחץ על איקון המחשב וגורר אותו באמצעות לחיצה ממושכת על העכבר ומשחרר את הלחיצה במקומות בו רוצה למקם את המחשב.

ב. אם מקום שחרור העכבר נמצא בתחום המפה:

(1) עדכן את מקום המחשב למיקום שחרור העכבר

ג. עדכן תצוגת המפה.

הרחבות : אין.

15.12.4 UC-4 - מכירת מחשב מכיתת מחשב

תיאור כללי: משתמש בעל הרשאה מתאימה יוכל למחוק מחשב מכיתת מחשב באמצעות בחירת איקון המחשב במפה ולהחיצה על מקש "Delete" או להחיצה על כפתור המחיקה.

angkanim: 用户拥有对地图上某台计算机的管理权限。

תנאי קדם: כיתת מחשב קיימת, המחשב הרצוי למכירה שייך לכיתת מחשב זו, למשתמש הרשות גישה מלאה/חלקית למחולקה אליה שייכת כיתת המחשב.

תנאי סיום: מחשב נמחק מכיתת המחשב.

תרחיש הצלחה עיקרי:

א. משתמש לוחץ על איקון המחשב אותו מעוניין ולוחץ על מקש "Delete" או על כפתור המחיקה.

ב. מחשב יוסר מרשימה המחשבים השביחים לכיתת מחשב זו.

ג. עדכן תצוגת המפה.

הרחבות : אין.

15.12.5. UC-5 - הוספת כיתת מחשב

תיאור כללי: משתמש בעל הרשאה מותאמת יוכל להוסיף כיתת מחשב חדשה באמצעות מילוי הפרטים הנדרשים.

조건ים: משתמש בעל הרשאה חלקית/מלאה.

תנאי קדם: אין.

תנאי סיום: נוצרת כיתת מחשב חדשה.

תרחיש הצלחה עיקרי:

א. משתמש לוחץ על "הוספת כיתה חדשה"

ב. משתמש יזין את שם הכיתה.

ג. שיפת המחלקות אליהן יש להשתמש גישה חלקית/מלאה.

ד. משתמש יבחר את המחלקה אליה יש לשיקן את כיתת המחשב מתוך רשימת המחלקות שלו.

ה. משתמש ימוך את כיתת המחשב על מפת האוניברסיטה.

ו. מפה כיתת המחשב תוצג למשתמש.

ז. שיפת המחשבים שאינם שייכים לאף כיתת מחשב ותציגם למשתמש.

ח. עברו כל מחשב שהמשמש בוחר מתוך רשימה זו :

(1) ביצוע UC-2 - הוספת מחשב לכיתת מחשב.

הרחבות: אין.

15.12.6. UC-6 - מחיקת כיתת מחשב

תיאור כללי: משתמש בעל הרשאה מותאמת יוכל למחוק כיתת מחשב מהמערכת באמצעות כפוץ מחיקת הכיתה.

조건ים: משתמש בעל הרשאה מלאה.

תנאי קדם: כיתת מחשב קיימת, למשתמש הרשאת גישה מלאה למחלקה אליה שייכת כיתת המחשב, משתמש לחץ "מחיקת כיתה" וקיבל אזהרה לגבי השלכות מחיקת הכיתה ובחר להמשיך בתהליך.

תנאי סיום: כיתת המחשב נמחקת.

תרחיש הצלחה עיקרי:

א. עברו כל מחשב השיקן לכיתת מחשב זו :

(1) מחק את המחשב מרישימת המחשבים של כיתת מחשב זו.

ב. עברו המחלקה אליה שייכת כיתת המחשב- מחק את כיתת מחשב זו מרישימת הכיתות השייכות למחלקה.

הרחבות: אין.

15.12.7. UC-7 - יצירת דוח

תיאור כללי: משתמש בעל הרשאה מותאמת שירצה להפיק דוח אודוטות כיתת/כיתות מחשב- יזין את הפרמטרים המבוקשים והמערכת תציג לו את הדוח הרצוי.

조건ים: משתמש בעל הרשאה צפיה/חלקית/מלאה.

תנאי קדם: אין.

תנאי סיום: תציגת הדוח המופק על המסך.

תרחיש הצלחה עיקרי:

א. משתמש יזין את טווח התאריכים וטוווח השעות הרצוי לדוח.

ב. שיפת כיתות המחשב הנמצאות במחלקות שימוש יש הרשאה כלשי אליהן ותציגם ברשימה.

ג. משתמש יבחר מתוך רשימת הכיתות את הכיתות שירצה לכלול בדוח.

ד. המערכת תפיק דוח בהתאם לפרמטרים שהניס המשמש.

ה. תצוגת הדוח למשתמש.

הרחבות : אין.

UC-8. 15.12.8 - תצוגת התראות

- תיאור כללי : משתמש בעל הרשאה מותאמת יכול לצפות בהתראות על אירועים שהגדר כחריגים.
- פתרונות : משתמש בעל הרשאה חלקית/מלאה.
- תנאי קדם : אין.
- תנאי סיום : תצוגת התראות.
- תרחיש הצלחה עיקרי :
- עברור כל מחלקה של משתמש יש הרשאה חלקית/מלאה :
 - עברור כל כיתת מחשב במחלקה :
 - עברור כל מחשב בכיתה מחושב :
 - אם מחשב עומד באחד הקרייטריונים שאוטם הגדר המשמש כאירוע חריג :
 - צור אירוע המכיל את שם המחשב, כיתה המחשב, המחלקה והקרייטריון חריג.
 - הוסף את האירוע לרשימה האירועים החריגים של המשתמש.
- הרחבות : אין.

UC-9. 15.12.9 - ניהול מדיניות התראות

- תיאור כללי : משתמש בעל הרשאה מותאמת יכול לעורוך את מדיניות ההתראות שלו : קבוע את הקרייטריונים לאירוע חריג ואת תכיפות קבלת ההתראות למילימ.
- פתרונות : משתמש בעל הרשאה חלקית/מלאה.
- תנאי קדם : אין.
- תנאי סיום : מדיניות התראות של המשתמש מעודכנת.
- תרחיש הצלחה עיקרי :
- משתמש יבחר לאחר כמה ימים של חוסר תגובה מחשב זה יחשב כאירוע חריג.
 - משתמש יבחר לאחר כמה ימים של חוסר התחברות משתמשים מחשב זה יחשב כאירוע חריג.
 - משתמש יבחר את תזרירות קבלת ההתראות למיליל : יומי/שבועי/חודשי או כלל לא.
 - הגדרות המשתמש ישמרו במערכת.
- הרחבות : אין.

UC-10. 15.12.10 - הוספת הרשות גישה

- תיאור כללי : משתמש בעל הרשאה מלאה יכול להוסיף למשתמש אחר הרשות גישה למחלקה מכל סוג.
- פתרונות : משתמש בעל הרשאה מלאה.
- תנאי קדם : המשתמש הרצוי להוספת גישה קיים במערכת ואין לו הרשות גישה למחלקה זו, למשתמש המוסיף הרשות גישה יש הרשות גישה מלאה למחלקה זו.
- תנאי סיום : נוספה הרשות גישה חדשה למשתמש הרצוי למחלקה זו.
- תרחיש הצלחה עיקרי :
- המשתמש בוחר את המחלקה אליה הוא רוצה לתת גישה מותוך רשימת המחלקות אליו יש לו הרשות גישה מלאה.
 - המשתמש בוחר את המשתמש לו רוצה לתת הרשות גישה.
 - המשתמש בוחר את סוג הרשות הגישה : צפיה/חלקית/מלאה.

ד. הרשות הגישה של המשתמש למחלקה נשמרת.

הרחבות: אין.

15.12.11. UC-11 - הורצת הרשות גישה

תיאור כללי: משתמש בעל הרשות מלאה יוכל למחוק למשתמש אחר הרשות גישה למחלקה.

শחקנים: משתמש בעל הרשות מלאה.

תנאי קדם: קיימת הרשות גישה למחלקה הרצוי למשתמש הרצוי למחיקת גישה, למשתמש המוחק הרשות גישה יש הרשות גישה מלאה למחלקה זו.

תנאי סיום: למשתמש הרצוי אין הרשות גישה למחלקה זו.

תרחיש הצלחה עיקרי:

א. המשתמש בוחר את המחלקה ממנו הוא רוצה למחוק הרשות גישה מתוך רשימת המחלקות אליו.

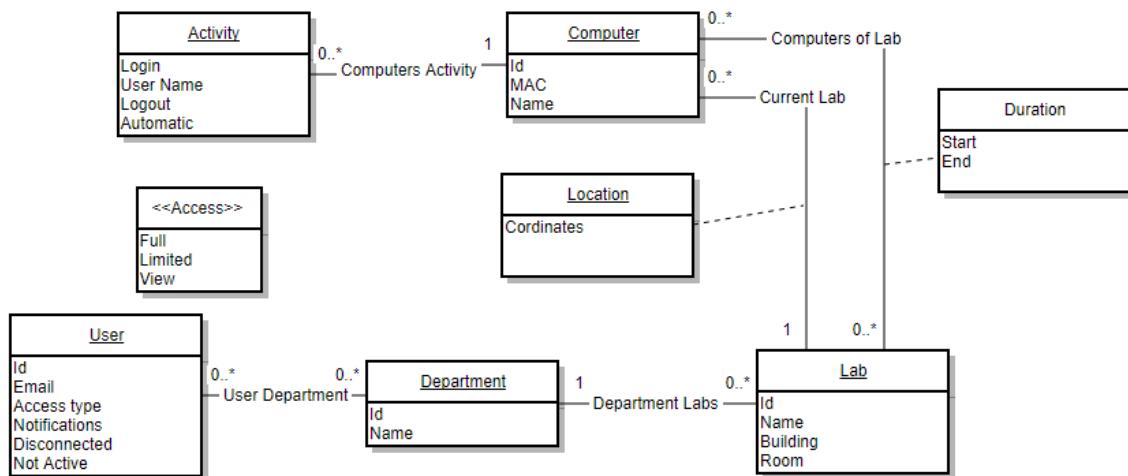
יש לו הרשות גישה מלאה.

ב. המשתמש בוחר את הרצוי להסרה מתוך רשימת המשתמשים בעלי הרשות כלשי במחלקה זו.

ג. הרשות הגישה של המשתמש הרצוי למחיקה למחלקה זו תוסר.

הרחבות: אין.

UML .15.3

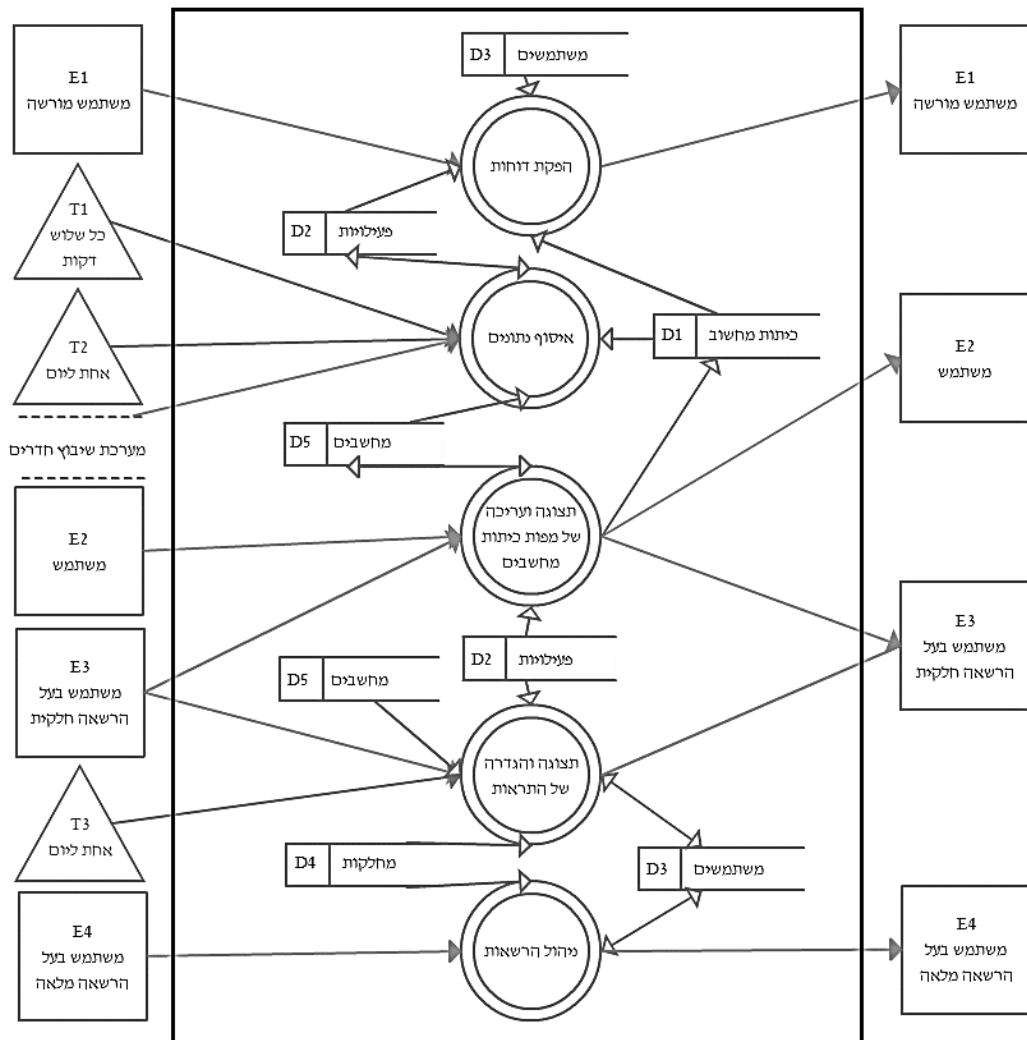


איור 4 - תרשימים מחלקות

טרנסקציות ותרשימי DFD .15.4

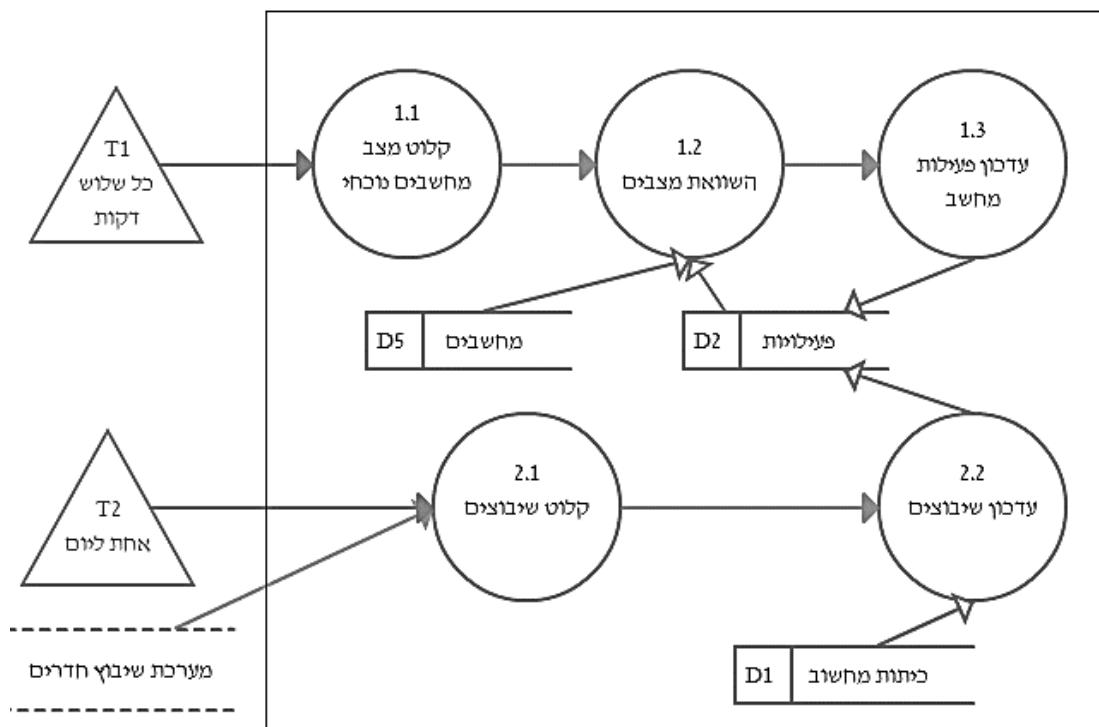
DFD-0 .15.4.1

איור 5 - תרשימים דרימת מידע ראשי



15.4.2. איסוף נתונים

איור 6 - DFD איסוף נתונים



א. נתוני שימוש

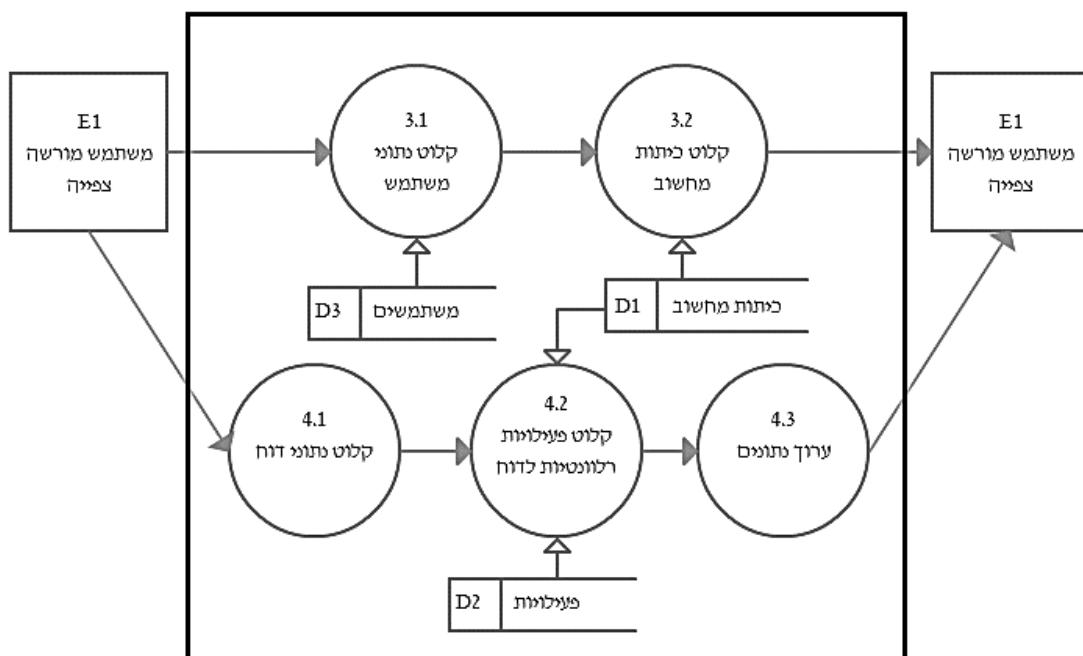
- (1) החלט טרנזקציה 1.1,1.2,1.3
- (2) קלוט מישות זמן אמיתי T1 : מצב מחשבים הנוכחי – שם מחשב, האם הוא דולק ושם משתמש מחובר (אם קיים אחרית NULL)
- (3) בצע פונקציה 1.1 : קלוט מצב מחשבים הנוכחי
- (4) קרא ממאגר D5 : ID – מזהה מחשב
- (5) קרא ממאגר D2 : פעילות – פעילות אחרת של כל מחשב
- (6) בצע פונקציה 1.2 : השוואת מצבאים (מעבר כל מחשב השוואת בין המצב הנוכחי לפעולות האחרונות)
- (7) בצע פונקציה 1.3 : עדכון פעילות מחשב
- (8) כתוב למאגר D2 : פעילות חדשה
- (9) סוף טרנזקציה

ב. שיבוץ חדרים

- (1) החלט טרנזקציה 2.1,2.2
- (2) קלוט מערכות שיבוץ חדרים : שיבוץ חדרים ליום הקרוב
- (3) בצע פונקציה 2.1 : קלוט שיבוץ חדרים
- (4) קרא ממאגר D1 : מזהה מחשבים שנמצאים בכיתות המחשב שקיים להם שיבוץ חדרים
- (5) בצע פונקציה 2.2 : עדכון שימושים
- (6) כתוב למאגר D2 : פעילות חדשה בשעות השיעור
- (7) סוף טרנזקציה

15.4.3. ניהול דוחות

איור 7 – FDF הפקת דוחות



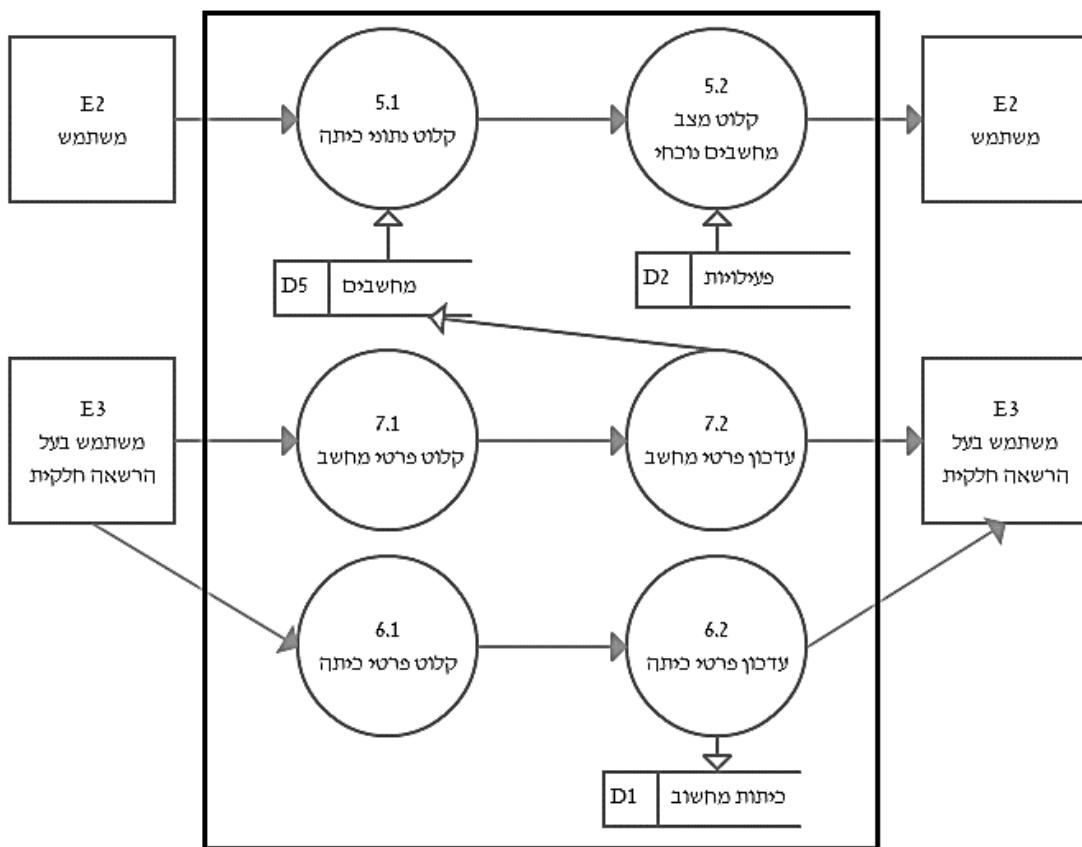
א. מלאי הנתונים

- (1) החל טרנסקציה 3.1,3.2
- (2) קלוט מישות משתמש E1 : מזהה משתמש
- (3) בצע פונקציה 3.1 : קלוט נתונים משתמש
- (4) קרא ממגר D3 : מחלקות שהמשתמש ממונה עליו
- (5) קרא ממגר D1 : כיתות מחשב
- (6) בצע פונקציה 3.2 : קלוט כיתות מחשב
- (7) הפק פלט לישות משתמש E1 : רשימת כיתות מחשב
- (8) סוף טרנסקציה

ב. הפקת הדוח

- (1) החל טרנסקציה 4.1,4.2,4.3
- (2) קלוט מישות משתמש E1 : נתונים דוח – תאריך התחלת, תאריך סיום, שעת התחלת, שעת סיום, רשימת כיתות מחשב.
- (3) בצע פונקציה 4.1 : קליטת נתונים דוח
- (4) קרא ממגר D1 : רשימת מחשבים
- (5) קרא ממגר D2 : פעילותות
- (6) בצע פונקציה 4.2 : קלוט פעילותות רלוונטיות לדוח
- (7) בצע פונקציה 4.3 : ערוך נתונים
- (8) הפק פלט לישות משתמש E1 : דוח.
- (9) סוף טרנסקציה

15.4.4. ניהול מפות



איור 8 - תצוגה וערכה של מפות ביצות מחשבים

א. **תצוגת ביצת מחשב**

- (1) החל טרנזקציה 5.1,5.2
- (2) קלוט מישות משתמש E2 : מזהה ביצה
- (3) קרא ממגר D5 : מחשבים בכיתת המחשב המבוקש
- (4) בצע פונקציה 5.1 : קלוט נתונים ביצה
- (5) קרא ממגר D2 : קרא פעילויות אחרונות
- (6) בצע פונקציה 5.2 : קלוט מצב מחשבים ווכח
- (7) הפק פلت לישות משתמש E2 : מפה של ביצת מחשב מבקשת
- (8) סוף טרנזקציה

ב. **הוספה/עריכה/מחיקה של ביצת מחשב**

- (1) החל טרנזקציה 6.1,6.2
- (2) קלוט מישות משתמש E3 : פרטי ביצה להוספה/עדכון/מחיקה – מספר בניין, שם הבניין ומספר הכיתה.
- (3) בצע פונקציה 6.1 : קלוט פרטי ביצה
- (4) בצע פונקציה 6.2 : עדכון פרטי ביצה
- (5) כתוב למאג'ר D1 : פרטי נתוני ביצה
- (6) הפק פلت לישות משתמש E3 : הודעה אישור
- (7) סוף טרנזקציה

*מחיקה תאפשר רק לישות משתמש בעלי הרשותה מלאה

ג. הוספה/עריכה/מחיקה של מחשב מכיתה מחשב

1. החל טרנסקציה 7.1,7.2

2. קלוט מישות משתמש E3 : פרטי מחשב להוספה/עדכון/מחיקה – מיקום בכיתה

3. בצע פונקציה 7.1 : קלוט פרטי מחשב

4. בצע פונקציה 7.2 : עדכון פרטי מחשב

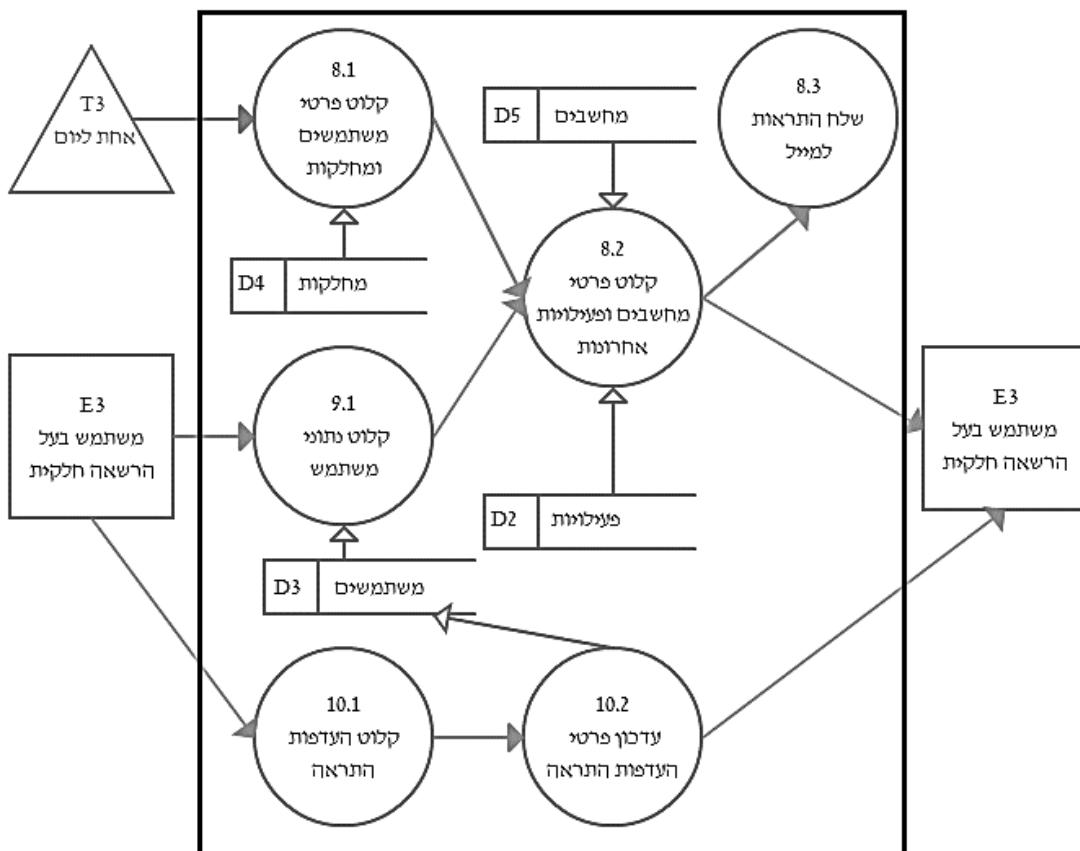
5. כתוב למאגר D5 : פרטי נתוני מחשב

6. הפק פלט לישות משתמש E3 : הودעת אישור

7. סוף טרנסקציה

15.4.5. ניהול התראות

איור 9 - DFD תצוגה והגדרה של התראות



א. שליחת התראה למיל

(1) החל טרנסקציה 8.1,8.2,8.3

(2) קלוט מישות זמן אמת T3 : משתמשים שהגדירו לקבל התראות למיל

(3) קרא ממאגר D4 : מחלקות של משתמשים אלו

(4) בצע פונקציה 8.1 : קלוט פרטי משתמשים ומחלקות

(5) קרא ממאגר D5 : מזהה מחשבים של מחלקות אלו

(6) קרא ממאגר D2 : פעילות

(7) בצע פונקציה 8.2 : קלוט פרטי מחשבים ופעילויות אחריות

(8) בצע פונקציה 8.3 : שלח מיל התראות

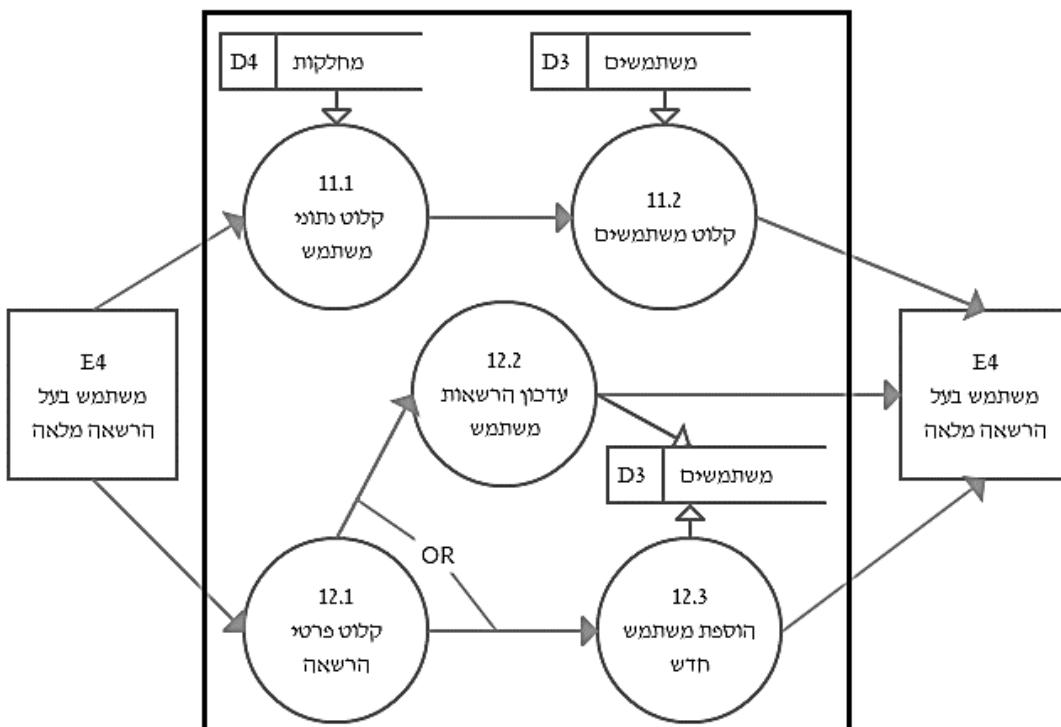
(9) סוף טרנסקציה

ב. תצוגת התראות במערכת:

- (1) החל טרנזקציה 9.1,8.2
 - (2) קלוט מישות משתמש E3 : מזהה משתמש
 - (3) קרא ממאגר D3 : מחלוקת של משתמש, העדפות התרבות
 - (4) בצע פונקציה 9.1 : קלוט נתונים משמש
 - (5) קרא ממאגר D5 : מזוהה מחשבים של מחלוקת אלו
 - (6) קרא ממאגר D2 : פעילות
 - (7) בצע פונקציה 8.2 : קלוט פרטי מחשבים ופעילויות אחרונות אחרונות
 - (8) הפך פלט לישות משתמש E3 : התרבות
 - (9) סוף טרנזקציה
- ג. **שינויי הגדרות של התרבות:**
- (1) החל טרנזקציה 10.1,10.2
 - (2) קלוט מישות משתמש E3 : העדפות התרבות
 - (3) בצע פונקציה 10.1 : קלוט העדפות התרבות
 - (4) בצע פונקציה 10.2 : עדכון פרטי העדפות התרבות
 - (5) כתוב למאגר D3 : פרטי נתונים העדפות התרבות
 - (6) הפך פלט לישות משתמש E3 : הודעה אישור
 - (7) סוף טרנזקציה

15.4.6. ניהול הרשות

איור 10 - DFD ניהול הרשות



א. **הציגת משתמשים:**

- (1) החל טרנזקציה 11.1,11.2
- (2) קלוט מישות משתמש E4 : מזהה משתמש

- ב. **שינוי הרשות משתמש או הוספת משתמש חדש**

 1. **החל טרנץקציה 12.1,12.2-12.3**
 2. **קלוט מישות משתמש E4 : פרטי הרשות – מידע מיידי (כפיה/חלקית/מלאה)**
 3. **בצע פונקציה 12.1 : קלוט פרטי הרשות**
 4. **אם <משתמש קיים> איזי :**

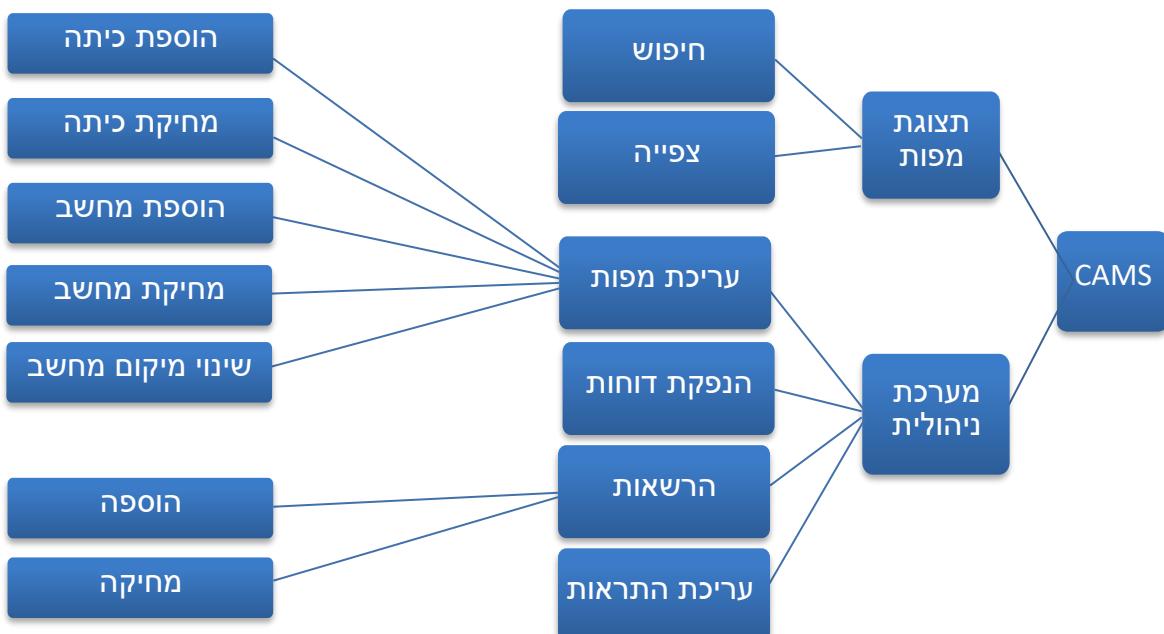
בצע פונקציה 12.2 : עדכון הרשות המשתמש

 5. **אחרת :**

בצע פונקציה 12.3 : הוספת משתמש חדש

 6. **כתוב למ Lager D3 : משתמש**
 7. **הפק פלט לישות משתמש E4 : אישור**
 8. **סוף טרנץקציה**

.15.5 עץ תפריטים



איור 11 - עץ תפריטים

16. סיכום עבודה ניתוח ועיצוב

חלק זה של העבודה עוסק בניתוח ועיצוב המערכת העתידית(החדשת) - ביצעו אפיון ראשי, הגדרת הדרישות, בניית הארכיטקטורה של המערכת ותרשים מחלקות. הצגנו את תרחישי השימוש של המערכת, הגדרנו את הטרנוזקיות הקיימות ואת זרימת המידע בין חלקיה.

כמו כן, ניתחנו את המערכת לפרטיו פרטים על מנת להבין בדיקות מה עלייה לעשוות ומה כולל בתחום האחריות המבצעית שלה.

שלב העיצוב הוא שלב קריטי במחזור חי המערכת ועל כן השקענו בו מחשבה רבה. נעשה ניתוח קפדי על מנת לצמצם ככל האפשר אי הבנות ותקלות שיכולות לקרות בעקבות כך בשלב מאוחר יותר. העיצוב נעשה תוך חישבות בייעילות התהליכים במטרה לשפר את הביצועים של המערכת. בשיקולים בנושא בניית בסיס הנתונים שמנוע דגש על ההתאמה לתהליכי הפעלים במערכת. סיום שלב זה מאפשר לנו לגשת לשלב הבא שהוא מימוש התוכנה.

17. מימוש המערכת

לאחר אפיון ועיצוב המערכת בוצע, בחלק זה של העבודה נתאר את שלב יישום המערכת החדשה. נציג את התוכנה, את חלקייה השונים, בסיס הנתונים, תהליכי עיקריים בלוגיקת המערכת, אבטחת מידע, הבדיקות והניסויים שביצענו, התאוששות מתקלות ומבנה הקבצים והתיקיות. בנוסף נציג צילומי מסך של המערכת ואת אופן הטעמאות המערכת. לבסוף נסכם את הפרויקט ואת התהילה שעברנו במהלך ביצוע הפרויקט השנה الأخيرة.

18. התוכנה

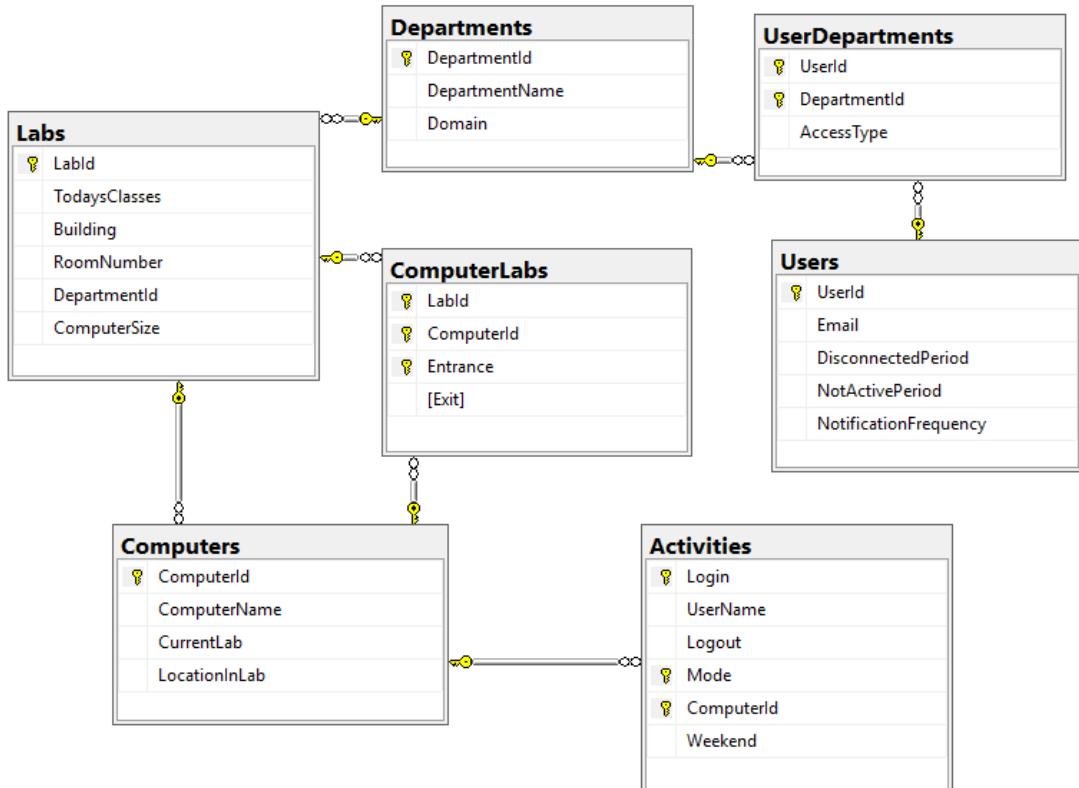
להלן פירוט המערכות, סביבות הפיתוח, סביבות העבודה, כל פיתוח העיקריים ששימשו אותנו בתהליך העבודה.

שם	מהדרגה
Microsoft Visual studio 2017	15.6.2
Entity Framework	6.2.0
jQuery	3.3.1
jQuery UI	1.12.1
Microsoft Asp.Net Mvc	5.2.6
Bootstrap	4.1.1
Microsoft Windows Server	2016
Microsoft PowerShell	5.1.15063.966
Microsoft SQL Server	Standard 64-bit 2017
Internet Information Services (IIS)	10

טבלה 5 - פירוט גרסאות

18.1. בסיס הנתונים

א. סכמת בסיס הנתונים :



איור 12 - סכמת בסיס הנתונים

ב. פירוט הטבלאות והשדות:
ב.1. טבלת Activities

טבלה זו מכילה את פעילותות המחשבים.

הסבר	מפתח זר	מפתח ראשי	סוג	שם שדה
זמן תחילת הפעולות		כן	DateTime	Login
שם המשתמש שמחובר למחשב, במידה ומחובר משתמש למחשב – אחרת NULL			string	UserName
זמן סיום הפעולות – במידה והפעולות עדין מתקיימת זמן הסיום הוא NULL			DateTime	Logout
סוג הפעולות (0 – משתמש מחובר למחשב, 1 – פעילות של שעור, 2 – מחשב ללא תקשורת)		כן	byte	Mode
מזהה המחשב	כן	כן	int	ComputerId
האם הפעולות התרחשה בסוף שבוע (יום שישי או שבת)			bool	Weekend

טבלה 6 - טבלת Activities

ג.1. טבלת Computers

טבלה זו מכילה את המחשבים.

שם שדה	סוג	מפתח ראשי	מפתח זר	הסבר
ComputerId	int	כן		מזהה המחשב
ComputerName	string			שם המחשב
CurrentLab	int	כן		מזהה כתובת המחשב שמהחישב נמצא בה בזמן הנוכחי – לquo מטבלת כתותות מחושב, אם איןנו נמצא בכיתת מחושב בזמן הנוכחי ערך זה יהיה NULL
LocationInLab	string			מקום המחשב בכיתות המחשוב הנוכחי – המרחק באחוזים של המחשב מהפינה השמאליתعلילונה של האובייקט שמכיל את המחשבים בתצוגה של התפוסה הנוכחי וערכית כיתה. בפורמט : "top,left" לדוגמא : "20%,40%" – כלומר המחשב נמצא במרחב 20% מהגבול העליון ו40% מהגבול השמאלי.

טבלה 7 - טבלת Computers

ד. Labs טבלת

טבלה זו מכילה את כוונות המחשב.

שם שדה	סוג	רפואי ראשי	מפתח זר	הסביר
LabId	int	כן		מזהה כיתת המחשב
TodaysClasses	string			שעות השיעורים שמתוכננים להיוון באותו הכיתה. בפורמט : "hh:mm-mm:hh:mm,hh:mm-08:00-12:00,13:30". לדוגמה : "hh:mm-00:00-11:00,12:00-13:30" – כלומר היום יש בכיתה המצווב שיעור בין השעות 8 בבוקר עד 11 ובין השעות 12 בצהרים ועד אחת וחצי.
Building	string			שם הבניין
RoomNumber	string			שם/מספר הכיתה
DepartmentId	int	כן		מזהה המחלקה – לקוח מטבלת מחלקות
ComputerSize	int			גודל (בפיקסלים) האיקונים של המחשבים בכיתת המחשב

טבלה 8 - טבלת Labs

ה. טבלת ComputerLabs

טבלה זו מכילה את תקופות השהות של מחשבים בכיתות המחשבוב.

שם שדה	סוג	מפתח ראשי	מפתח זר	הסבר
LabId	int	כן	כן	מזהה כיתת המחשב
ComputerId	int	כן	כן	מזהה מחשב – לקוח מטבלת מחשבים

זמן הכניסה של המחשב לכיתת המחשב		כן	DateTime	Entrance
זמן החיצאה של המחשב מabitatת המחשב			DateTime	Exit

טבלה 9 - טבלה 9

1.1 טבלת Departments :

טבלה זו מכילה את המחלקות.

שם שדה	סוג	מפתח ראשי	מפתח זר	הסבר
DepartmentId	int	כן		מזהה המחלקה
DepartmentName	string			שם המחלקה
Domain	string			שם הדומיין שמחברי המחלקה נמצאים בו

טבלה 10 - טבלה 10

1.2 טבלת UserDepartments :

טבלה זו מכילה את ההרשאות שיש למשתמשים על מחלקות.

שם שדה	סוג	מפתח ראשי	מפתח זר	הסבר
UserId	int	כן	כן	מזהה משתמש – לקוח מטבלה משתמשים
DepartmentId	int	כן	כן	מזהה מחלקה – לקוח מטבלה מחלקות
AccessType	byte			סוג הגישה שיש למשתמש על כיתות המחשב של המחלקה (0 – הרשאה מלאה, 1 – הרשאה חלקית, 2 – הרשות צפיה)

טבלה 11 - טבלה 11

1.3 טבלת Users :

טבלה זו מכילה את המשתמשים.

שם שדה	סוג	מפתח ראשי	מפתח זר	הסבר
UserId	int	כן		מזהה משתמש
Email	string			דוא"ל של המשתמש, משמש לשילוח התראות לדוא"ל ולהזדהות במערכת
DisconnectedPeriod	int			לאחר כמה ימים ברצף שבו המחשב לא תקשרוות המשתמש יקבל התראה על כך
NotActivePeriod	int			לאחר כמה ימים ברצף שבו לא היה המשתמש מחובר למחשב המשתמש יקבל התראה על כך
NotificationFrequency	byte			באייזו תדירות המשתמש יקבל התראות לדוא"ל (0 – בכלל לא, 1 – אחת ליום, 2 – אחת לשבוע, 3 – פעם בחודש)

טבלה 12 - טבלה 12

18.2. תħallid aissuf natvunim

א. על מנת לעקוב אחר פעילות המחשבים בכיתות הציבוריות, לאפשר הצגה של מצב תפוצה נוכחת של כיתה ולהפיק דוחות- היה علينا לאסוף נתונים על פעילות המחשבים. כל שלוש דקוטה המערכת בודקת את מצב כל המחשבים בכל כיתות המחשב על מנת לעדכן את טבלת הפעולות שלהם בסיס הנתונים. עבור כל מחשב נבדק האם הוא דלוק, האם יש משתמש המחבר אליו ומהו שם המשתמש שלו. לפי טבלת הפעולות הנוכחית שלו, היה מצב כל מחשב בפעם האחרונה שבדקנו על ידי שליפת הפעולות הנוכחית שלו.

ב. המצביעים האפשריים של פעילות מחשב:

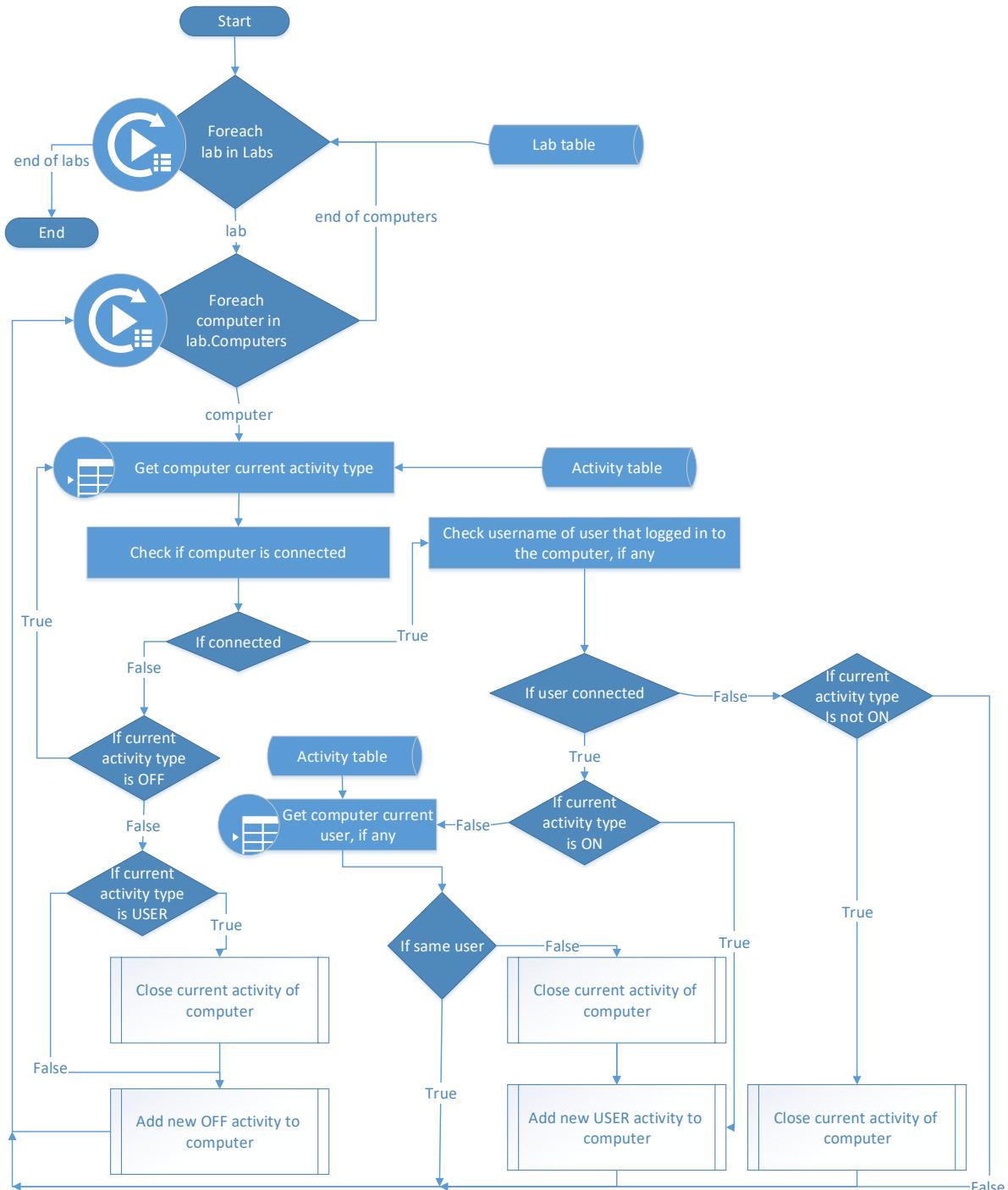
OFF- כבוי או אין מגיב.
USER- משתמש מחובר.
ON- דлок ופנוי- זהו המצב התדייר ביותר ובחרנו שלא לשמר רשומות עבورو בטבלת הפעולות על מנת לא להעיסס על בסיס הנתונים. במקום זה אנחנו מסיקים עבור כל זמן שעבورو לא מתועדת פעילות שהמחשב נמצא במצב ON.

CLASS- מועבר שיעור בכיתה בה נמצא המחשב- מצב זה אינו רלוונטי כאן כי אנו אוספים את מצב פעילות המחשבים בפועל ולכן אנו מתעלמים מרשות אל. הן משתמשות אותו להפקת דוחות.

ג. מתבצעת השוואת בין מצב המחשב בפעם האחרונה שבדקנו(מצביע בסיס הנתונים) לבין מצבו הנוכחי שקיבלו עכשו:

- 1) אם מחשב היה במצב "ON" וכעת הוא מנותק- תיפתח פעולה חדשה של מצב "OFF" שהתחילה בזמן הנוכחי.
- 2) אם מחשב היה במצב "ON" וכעת משתמש מחובר אליו- תיפתח פעולה חדשה של מצב "USER" עם שם המשתמש שהתחילה בזמן הנוכחי.
- 3) אם מחשב היה במצב "USER" וכעת מחובר אליו משתמש אחר- נעדכן את זמן הסיום של פעולה "USER" הקודמת לזמן הנוכחי ונפתח פעולה "USER" חדשה עבור המשתמש החדש.
- 4) אם מחשב היה במצב "USER" וכעת הוא דлок ופנוי- נעדכן את זמן הסיום של פעולה המשתמש הקודם לזמן הנוכחי.
- 5) אם מחשב היה במצב "OFF" וכעת הוא תפוס על ידי משתמש- נעדכן את זמן הסיום של פעולה "OFF" הקודמת לזמן הנוכחי ונפתח פעולה "USER" חדשה עבור המשתמש החדש.
- 6) אם מחשב היה במצב "OFF" וכעת הוא דлок ופנוי - נעדכן את זמן הסיום של פעולה הכבוי לזמן הנוכחי.

ד. תיאור לוגי:



תרשים 1 - תהליך איסוף נתונים

.18.3 תרחישי שימוש

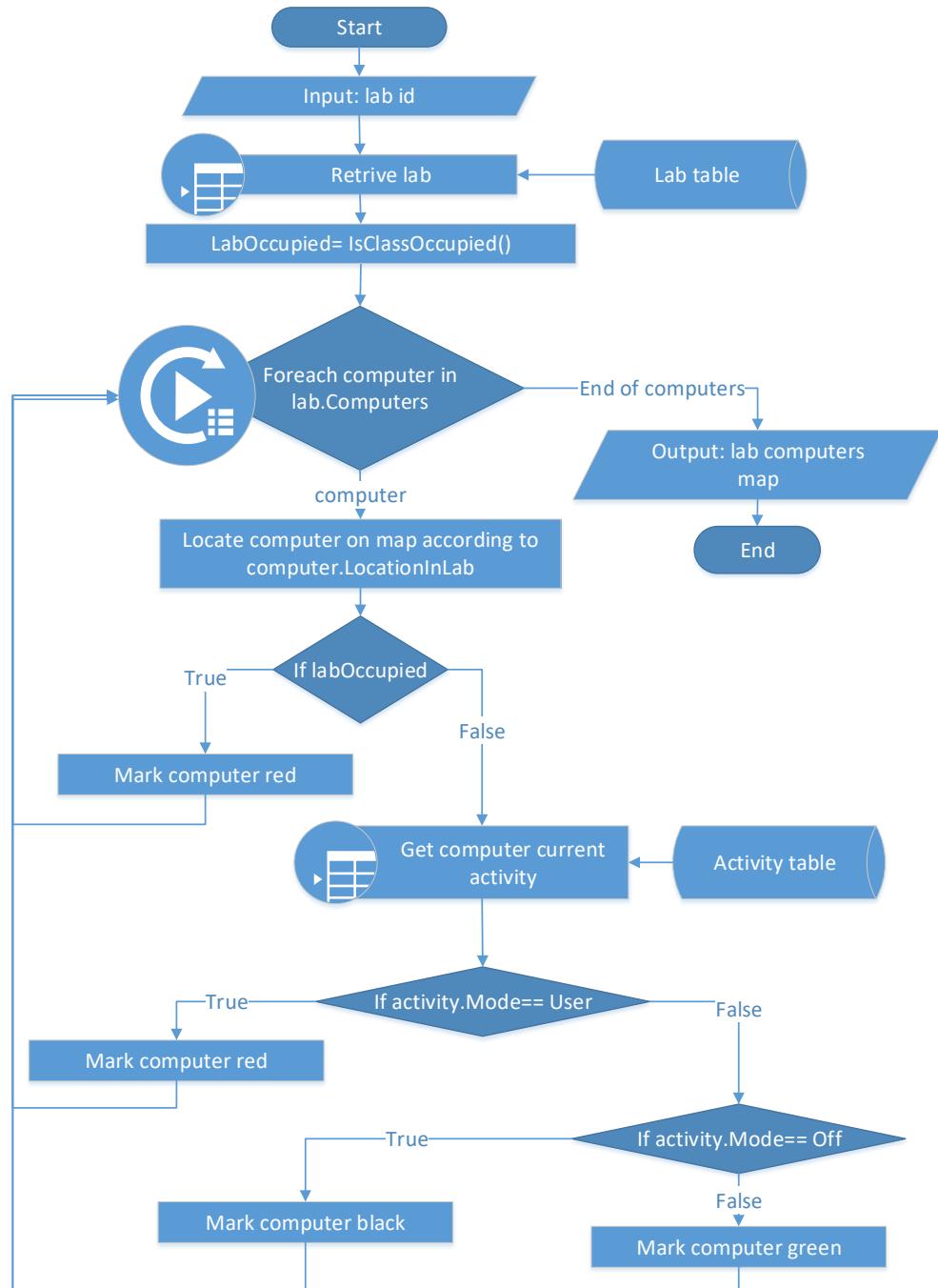
18.3.1. צפיה בכיתות מחובב - UC-1

א. **תיאור כללי:**

כל משתמש יוכל להיכנס למפת התפוצה הנוכחי של כויתת מחשב ציבורית ולראות את מצב המחשבים בה. המחשבים יהיו פזוריים בהתאם למיקומים הפיזיים בគיתה כפי שהוגדר בעת יצירת המחשבים בה.

ויריכת הכיתה. צבע כל מחשב ייעד על מצבו- מחשב ירוק מסמל מחשב פנוי, מחשב אדום מסמל מחשב תפוס ומחשב שחור מעיד על מחשב כבוי (או תקול). קודם כל מתבצע בדיקה אם מתקיים שיעור בכיתת המחשב בנקודת הזמן הנוכחיית- אם כן יסמננו כל המחשבים כתפוסים, במידה ולא יבדק מצב כל מחשב בהתאם לפעולות הנוכחות שלו בטבלת הפעולות.

ב. תיאור לוגי:



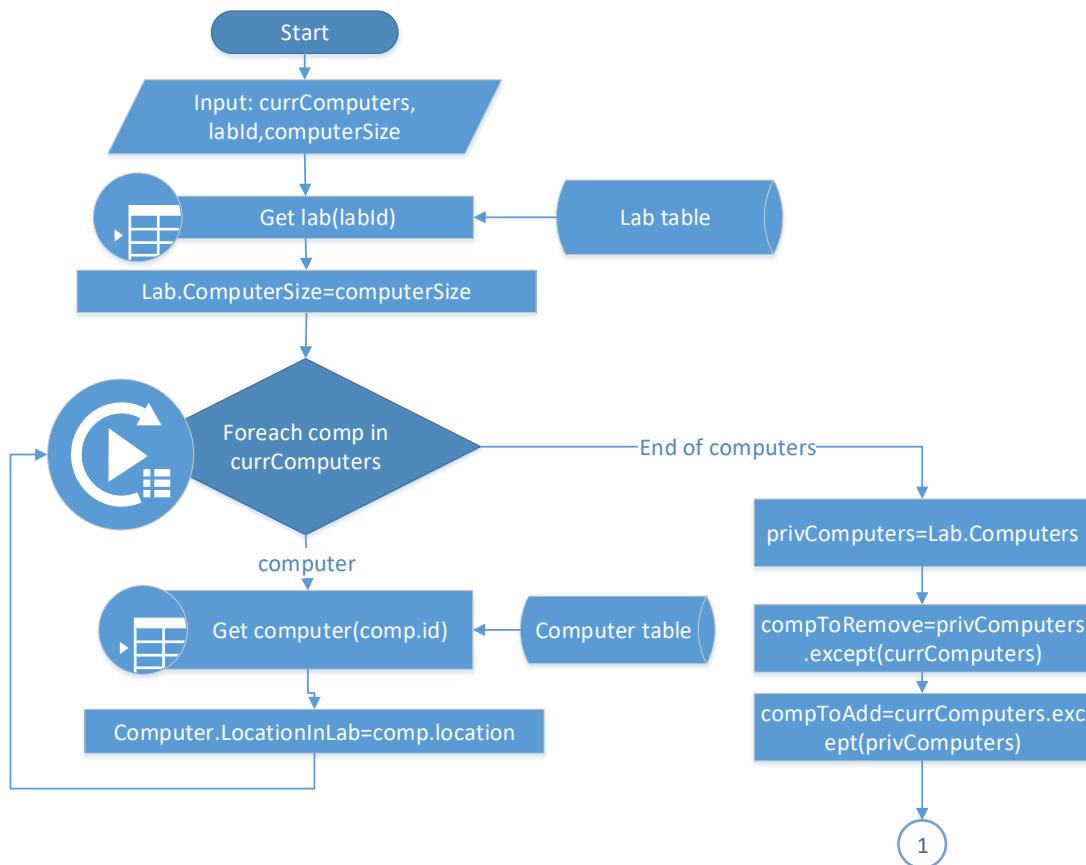
תרשים 2 – צפיה בכיתות מחשב

18.3.2. ערכות כיתת מחשב UC-2,3,4

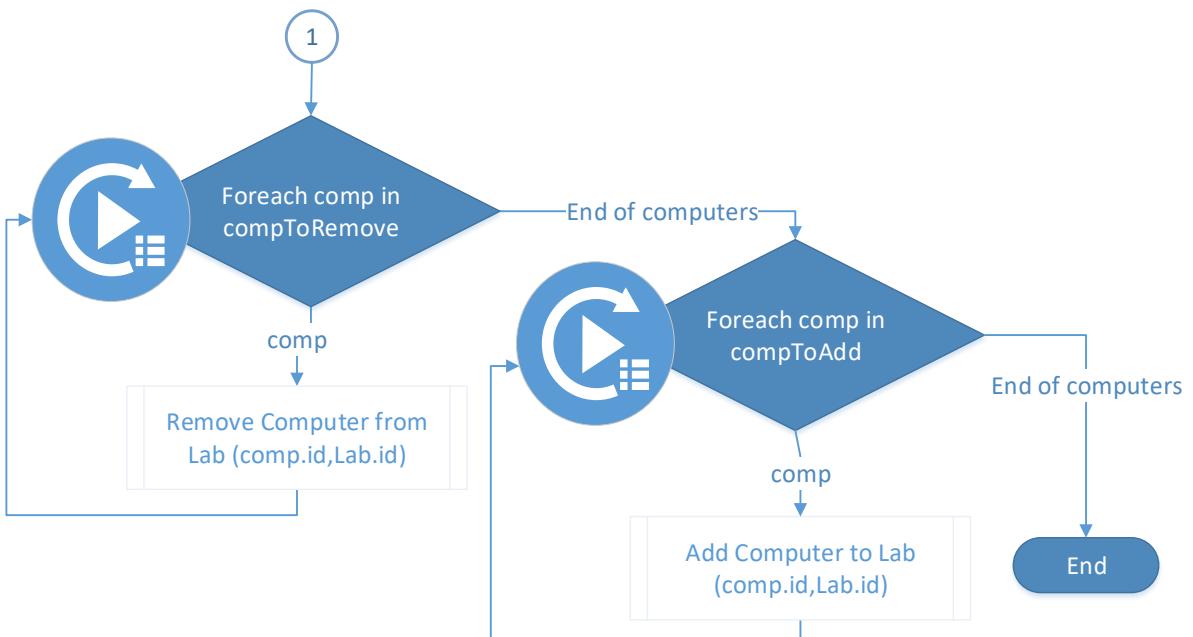
א. **תיאור כללי:**

- 1) הרשות גישה חלקית ומלאה למחלקהאפשרת למשתמש המחזיק בה לעורך כיתות מחשוב השויות למחלקה זו. בעת כניסה המשתמש במצב עריכת כיתה הוא יראה את מיפוי המחשבים הנמצאים בכיתה זו כאשר כל מחשב מיוצג על ידי איקון של מחשב. ניתן לבחור מה יהיה גודל האיקונים של המחשבים כדי להתאים את המפה לגודלים שונים של כיתות (ישן כיתות מחשוב ברוחבי האוניברסיטה המכילות מחשבים בודדים אך קיימות כיתות המכילות עשרות מחשבים). קיימים שלושה גודלים של איקונים מותאם המשמש יכול לבחור.
2. **הוספת מחשב לכיתת מחשוב-** בצד המסך יראה המשתמש את כל המחשבים אותם הוא יכול להוסיף לכיתה (מחשבים השוכנים לדומיין של כיתה זו אשר אינם שייכים לאף כיתת מחשוב אחרת) אותם הוא יכול לגרור למפה ולמקסם כרצונו.
- ii. **מחיקת מחשב-** ניתן למחוק מחשב מהכיתה על ידי סימון האיקונים שלהם על גבי המפה ולהזיה על כפטור המחיקה.
- iii. **שינוי מקום מחשב-** המשתמש יכול להזיה את המחשבים הפזורים על גבי המפה על ידי גירירת האיקונים שלהם למקום הרצוי.
- 2) כאשר ילחץ המשתמש על שמור, מצבה המעודכן של הכיתה- כל המחשבים ומיקומם ישלו לשרת אשר יעדכן את בסיס הנתונים בהתאם. מתבצעת השוואה בין מצב הכיתה לפניה העריכה ואחריה: עברו כל מחשב יעדכן מיקומו בכיתה מחשוב בסיס הנתונים בטבלת Computers.
- 3) עברו מחשב שלא היה בכיתה מחשוב ונוסף- מעודכנים את הכיתה הנוכחי של המחשב להיות כיתת המחשב שנערכה ומתחדדים בטבלת ComputerLab את כניסה המחשב לכיתה זו.
- 4) עברו מחשב שהיה בכיתה מחשוב והוסר- מעודכנים כי אין כיתה נוכחית למחשב זה ומעודכנים בטבלת ComputerLab את תאריך יציאת המחשב מכיתה המחשב.

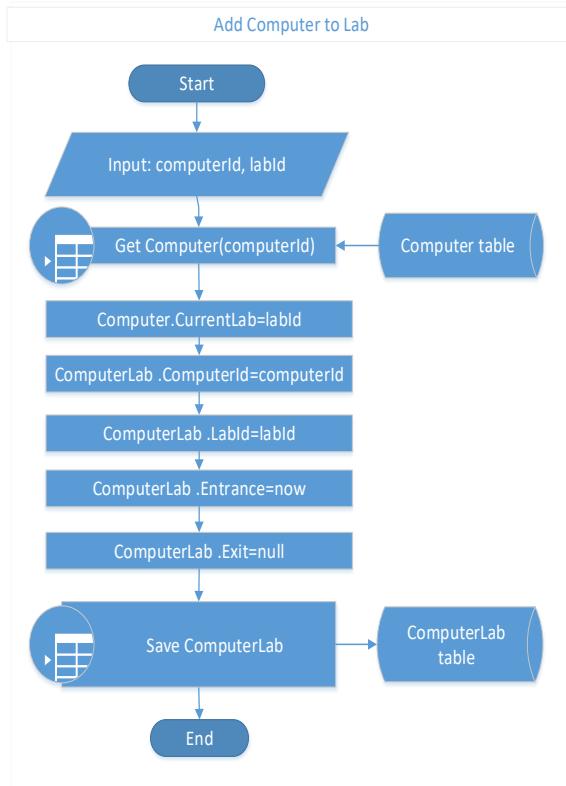
ב. תיאור לוגי:



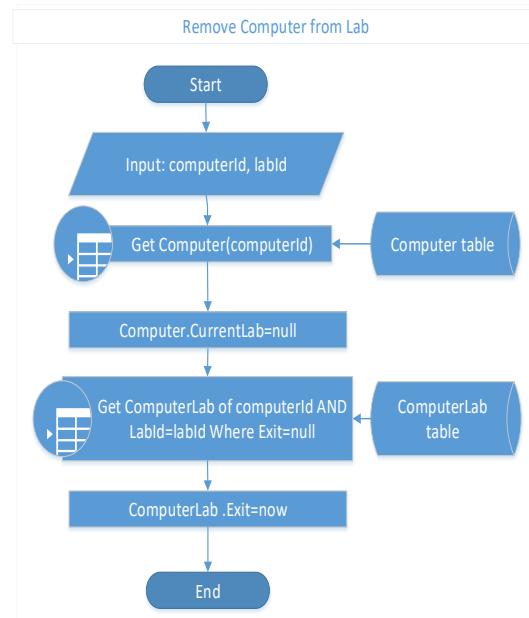
תרשים 3 - עיבת כיתת מחשב חלק א'



תרשים 4 - עיבת כיתת מחשב חלק ב'



תרשים 5 - הוספה מחשב לכיתת מחשב



תרשים 6 - מחיקת מחשב מכיתת מחשב

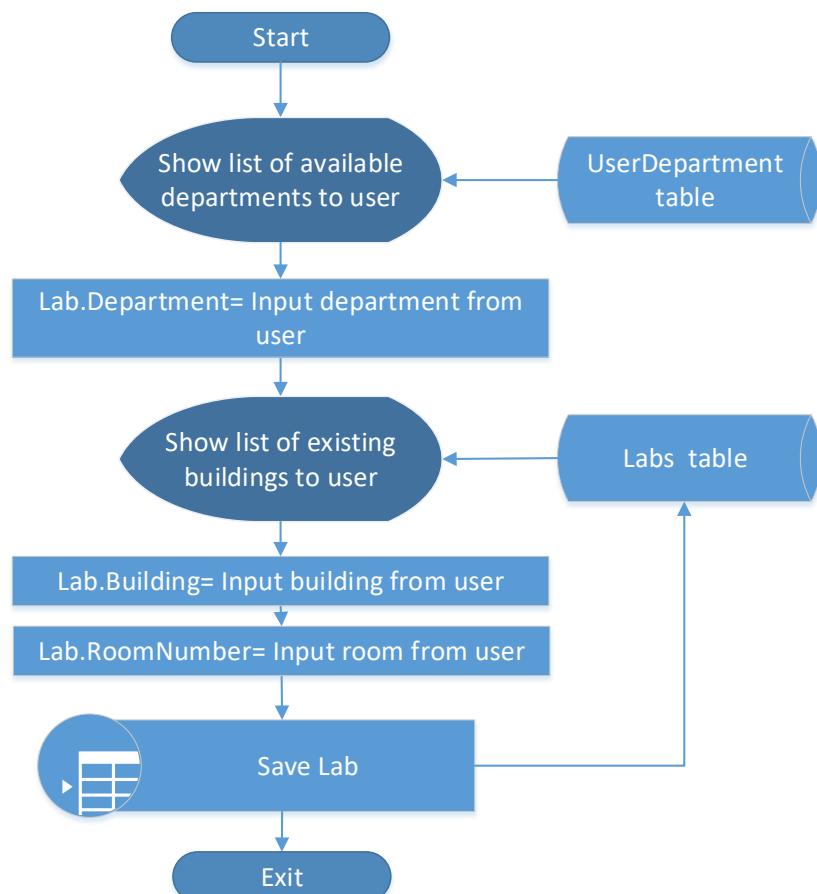
18.3.3. הוספת כיתת מחשב UC-5

א. **תיאור כללי:**

הרשאת גישה חלקית ומלאה למחלקה משתמש המחזיק בה להוסיף כיתות מחשב למחלקה זו.

כאשר יבחר משתמש ל揖צר כיתה חדשה הוא יבחר לאיזה מחלקה שייכת כיתה זו ומתוך רשימת המחלקות אליהן יש לו הרשאת גישה חלקית או מלאה, בנוסף יבחר באיזה בניין נמצא נמצאת הכתובת מתוך רשימת כל הבניינים הקיימים במערכת ולבסוף יזון את מספר הכתובת. בסיום מילוי השדות ושמירה הנתונים ישלוו לשרת והכיתה תוסף למערכת.

ב. **תיאור לוגי:**



תרשים 7 - הוספת כיתת מחשב

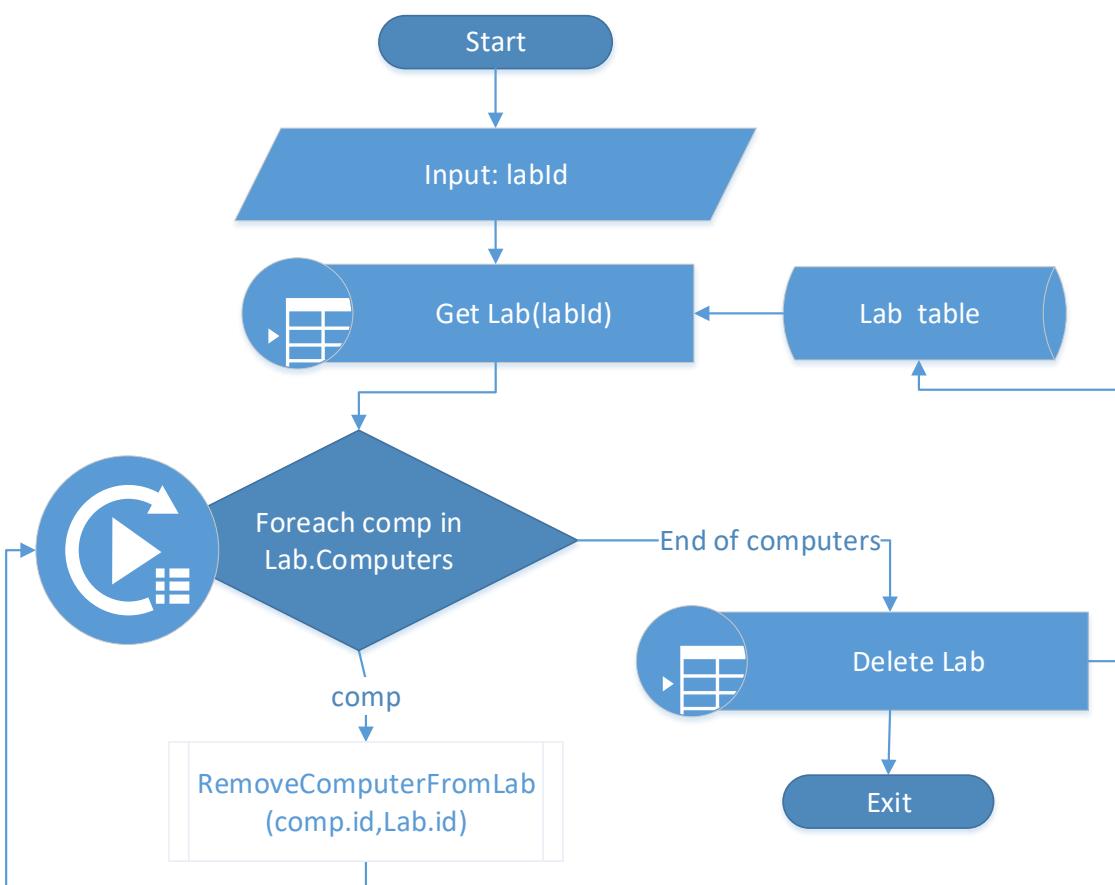
UC-6. מחיקת כיתת מחשב 18.3.4

א. תיאור כללי:

הראשת גישה מלאה למחלקה מאפשרת למשתמש המחזיק בה למחוק כיתות מחשב ממחלקה זו.

כאשר משתמש ימחק כיתה זו תימחק מבוסיס הנתונים ויתעדכן עבור כל המחשבים המשויכים לכיתת מחשב זו שאין להם כיתה נוכחית.

ב. תיאור לוגי:



תרשים 8 - מחיקת כיתת מחשב

18.3.5 UC-7. דוח יצירת

א. **תיאור כללי:**

משתמש יכול להפיק דוחות עבור כיתות מחשב שנמצאות במחולקות עboroן יש לו הרשות גישה כלשהי (צפיה/חלקית/מלאה). על מנת ליציר דוח יבחר המשתמש טווח תאריכים וטוווח שעות עליו יופק הדוח, ניתן להגדיר כי לא תהיה התחשבות בסופי שבוע (ימים אשר לא מוגדרים כימי עבודה ויכולים להשפיע על הסטטיסטיקה) ומהו הדוח הרצוי מתוך רשימה של דוחות אפשריים. המערכת תפיק את הדוח בהתאם לפרמטרים שהכניס המשתמש.

A.1. דוח אחוז שימוש ממוצע

נדיר:

- (1) **חומר התחשבות במערכת שיבוץ החדרים**- מתחשבים רק בשימוש של משתמשים במחשבים בכיתה.
- (2) **התחשבות במערכת שיבוץ החדרים**- מתחשבים בשימוש של משתמשים במחשבים בכיתה וגם בשיעורים שהתקיימו בכיתה. במקרה שיעור בכיתה כל המתחשבים בכיתה זו ייחסבו בשימוש גם אם בפועל אין משתמש מחובר למחשב.
- (3) **אחוז שימוש ממוצע של מחשב**- סך זמן השימוש במחשב בזמן נתון סך הזמן שהוא היה זמין בזמן זה :

$$\frac{\text{Total usage time}}{\text{Total available time}}$$

- (4) **אחוז שימוש ממוצע של כיתת מחשב**- סך זמן השימוש בכל המחשבים הנמצאים בכיתה במצטבר מתוך סך הזמן שמחשבים אלה היו זמינים בכיתה המחשב :

$$\frac{\sum_{\text{computers}} \text{Total usage time}}{\sum_{\text{computers}} \text{Total available time}}$$

דוח אחוז שימוש ממוצע מראה עבור כל כיתת מחשב מהו אחוז השימוש הממוצע בה עם ובלי התחשבות במערכת שיבוץ החדרים, ומאפשר גם להיכנס לפירוט העמדות של כל כיתת מחשב ולראות את אחוז השימוש הממוצע של כל מחשב בנפרד.

על מנת לחשב את סך זמן השימוש עבור כיתת מחשב علينا להתחשב בפעילויות כל המתחשבים אשר היו בכיתה מחשב זו **ב佗וח התאריכים הנוכחי**. בעזרת בטבלת ComputerLab המתעדת את תאריך הכניסה והיציאה של כל מחשב למעבדה אנו יודעים מהם המתחשבים שהיו בכיתה מחשב בטוח נתון.

אופן חישוב סך זמן השימוש במחשב :

נדיר : תאריך התחלה לדוח ,Report_Start - תאריך סוף לדוח ,Report_End -Enterance ,תאריך כניסה למחשב Exit . תאריך יציאת המחשב מכיתת המחשב.

עבור כל מחשב נבדק מה טווח התאריכים שעבורו יבדקו הפעילויות לצרכי הדוח - הטווח יהיה החפיפה בין טווח התאריכים שהוגדר לדוח לבין טווח התאריכים שהמחשב נמצא במעבده.

כלומר עבור כל מחשב יילקחו כל הפעילויות החל מ-*Report_Start, Enterance* ועד *.min(Report_End, Exit)*

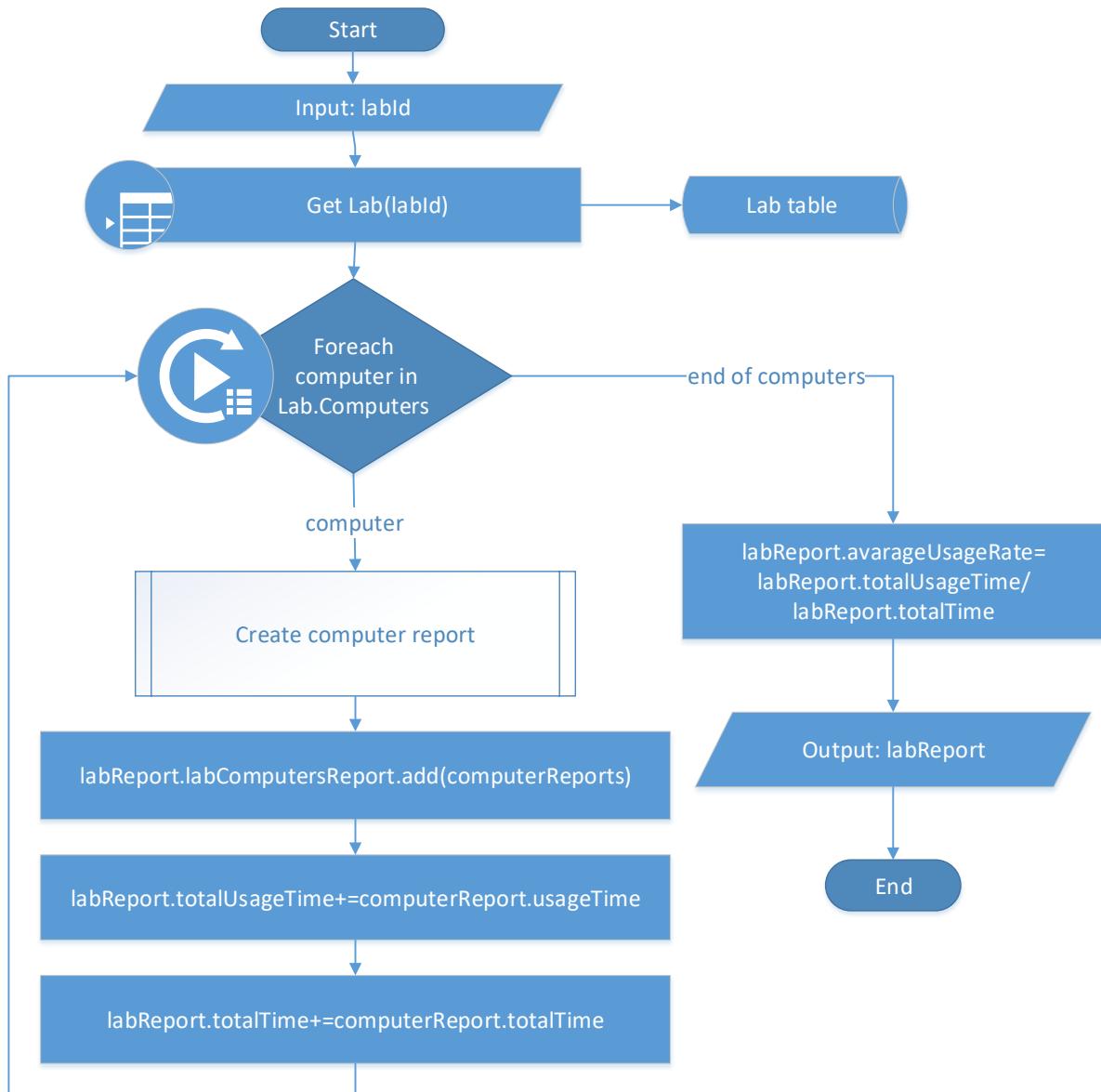
כדי לחשב את סך זמן השימוש במחשב עבור על כל הפעילויות שמצאו נוריד את הפעילויות המתועדות בסופי שבוע במידה והוגדר לא להתחשב ביום אלה. כיוון שהדוח מציג את אחוזי השימוש עם ובלי מערכת שיבוץ החדרים יתבצעו שני חישובים :

- i. אחוז שימוש ממוצע בלי התחשבות במערכת שיבוץ החדרים- מתוך כל הפעילויות המתועדות שמצאו נלקחות רק הפעילויות המוגדרות כפעולות משתמש (כל פעולה מתעדת את זמן ההתחברות וההנתתקות של משתמש למחשב).
- ii. אחוז שימוש ממוצע עם התחשבות במערכת שיבוץ החדרים- מתוך כל הפעילויות המתועדות שמצאו נלקחות הפעילויות המוגדרות כפעולות משתמש וכפעולות שיעור (זמן התחלה וסוף של שיעורים שהתקיימו בכיתת המחשב של המחשב).

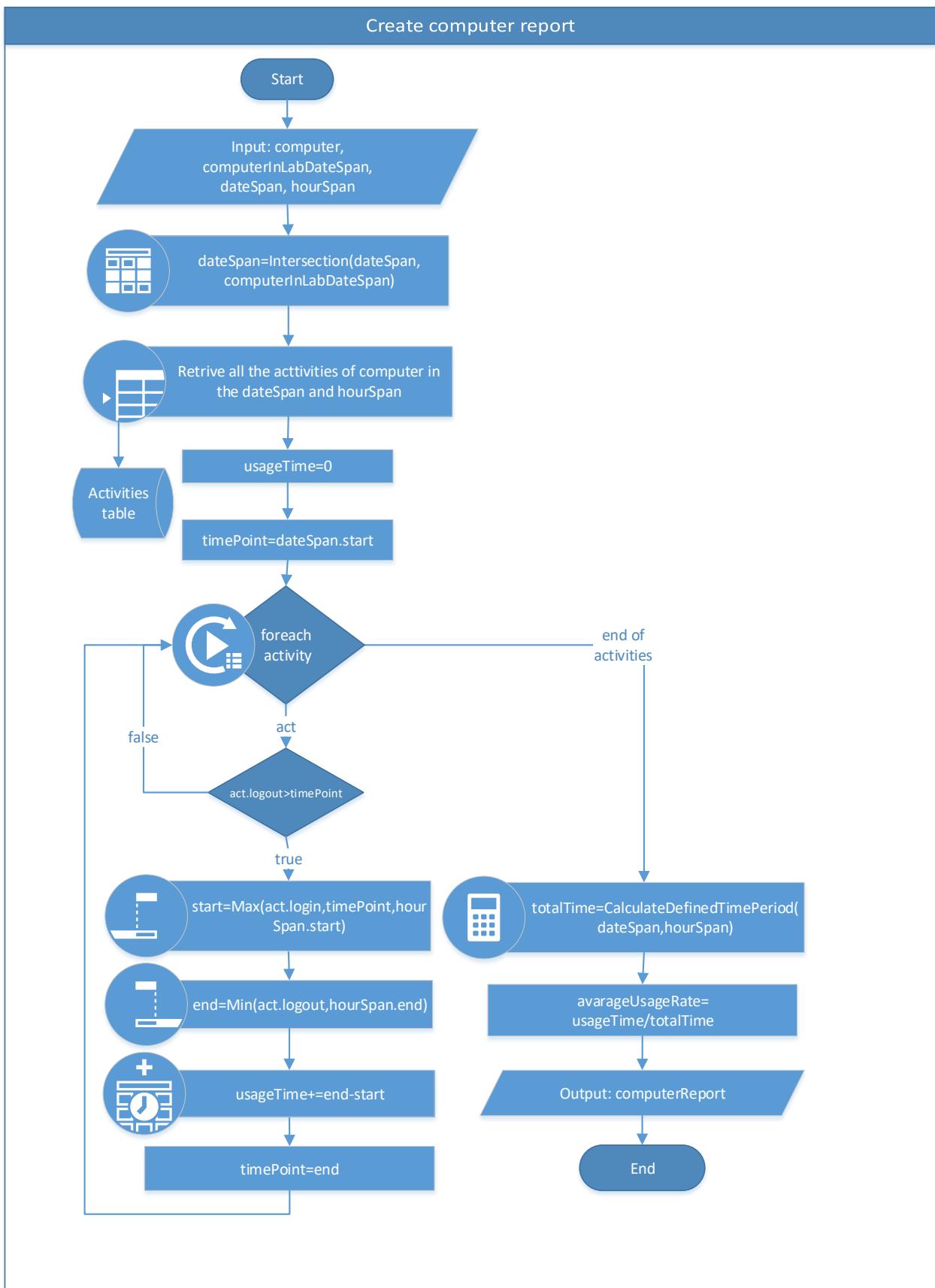
עבור כל פעילויות יילקח בחשבון רק זמן הפעילויות בטווח השעות שהוגדר לדוח ורק אם היא לא חופפת בזמן לפעילות אחרת שכבר חושבה (כאשר מתחשבים במערכת שיבוץ החדרים יכול להיווצר חפיפה בזמנים בין פעילות שיעור ופעולות משתמש). את כל הזמןים הללו נסכם לסך זמן השימוש במחשב, כאשר נחלק זמן זה בסך הזמן שהמחשב היה זמין בטווח הדוח ובכיתת המחשב נקבל את זמן השימוש הממוצע של המחשב.

לאחר חישוב זה עבור כל מasin בכיתת מחשב מtabצע החישוב עבור זמן השימוש הממוצע של כיתת המחשב.

א.2. **תיאור לוגי :**



תרשים 9 – יצירת דוח אחוז שימוש ממוצע



ב.1 דוח אחוז תפוסת מחשבים

נדיר :

- 1) אחוז מחשבים פעילים בכיתת מחשוב- כמה מחשבים נמצאים בשימוש מתוך סך המחשבים הנמצאים בכיתת המחשב בזמן נתון :

$$\frac{\# \text{Computers in use}}{\# \text{Total computers}}$$

דוח זה מציג את מצב אחוז תפוסת המחשבים בכיתת מחשוב לפי השעה ביום ולפי יום בשבוע.

עבור כל שעה ביום יוצג האחוז הממוצע של המחשבים שהיו פעילים בכיתת המחשב במהלך שעה זו בטוחה הזמנית שהוגדר לדוח, וכן כמה מה האחוז המקסימלי והמינימלי של המחשבים שהיו פעילים בשעה זו במהלך התקופה זו. בהגדרת הדוח משתמש בחורר את טווח התאריכים והשעות עליו ירצה לראות את הנתונים, עבור כל שעה בטוחה שהגדר: נסוף ונשמר ברשימה את אחוז המחשבים הפעילים בכיתת המחשב לכל יום בטוחה וכך נרצה להציג את הדוח למשתמש- נחשב את הממוצע, המקסימום והמינימום מתוך רשימה זו.

באופן דומה יוצג האחוז הממוצע, המקסימלי והמינימלי של המחשבים הפעילים בכיתת המחשב עבור כל יום בשבוע (ראשון- שבת). עבור כל יום בשבוע נסוף ונשמר ברשימה את האחוז המקסימלי של המחשבים הפעילים, המינימלי והממוצע של כל הימים בטוחה שהגדר המשתמש שלהם ביום השבוע המתאים.

האחוז המקסימלי/מינימלי של המחשבים הפעילים עבור יום אחד הוא האחוז המקסימלי/מינימלי מבין אחוזי המחשבים הפעילים שהתקבלו עבור כל שעה ביום זה. האחוז הממוצע של המחשבים הפעילים עבור יום הוא ממוצע האחוזים שהתקבלו לפי כל שעה ביום זה.

לדוגמא :

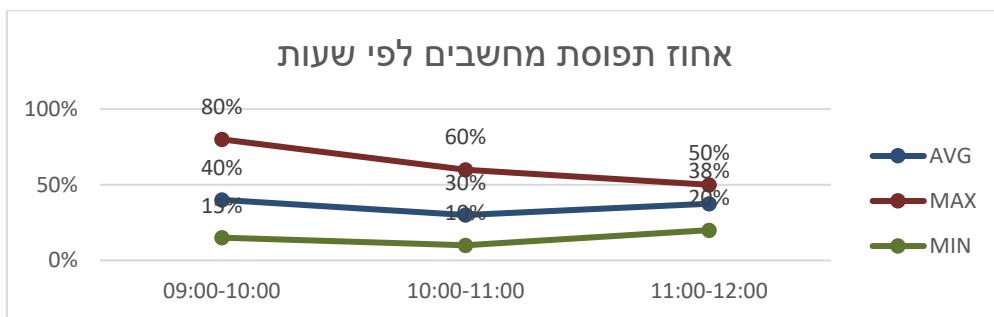
עבור דוח לכיתת מחשב מסוימת נרצה לקבל את אחוזי תפוסת המחשבים בטוחה התאריכים 3/6/18-10/6/18 בשעות 00:00-12:00, ללא התחשבות בסופי שבוע. נחשב את אחוז תפוסת המחשבים הפעילים בכיתת המחשב עבור כל שעה בכל יום בטוחה ונקבל :

10/6 (ראשון)	7/6 (חמישי)	6/6 (רביעי)	5/6 (שלישי)	4/6 (שני)	3/6 (ראשון)	
80%	40%	35%	50%	20%	15%	09:00-10:00
20%	10%	50%	60%	20%	20%	10:00-11:00
35%	40%	50%	40%	20%	40%	11:00-12:00

כיוון שהוגדר כי אין להתחשב בסופי שבוע, תאריכים 6/9-8 לא נלקחו בחשבון. עבור כל שעה בכל יום נבדק ונשמר מה אחוז המחשבים בשימוש.Cutת נוכל לחשב את הממוצע, המקסימום

והמיןימום עבור כל שעה ב佗וח וכל יום בשבוע. נניח עבור השעה 00:10-00:09 - אחוז השימוש המקסימלי הוא 80%, המינימלי 15%, והממוצע הוא 40%.

דו"ח אחוז תפוזת המחשבים לפי שעות יראה כך :

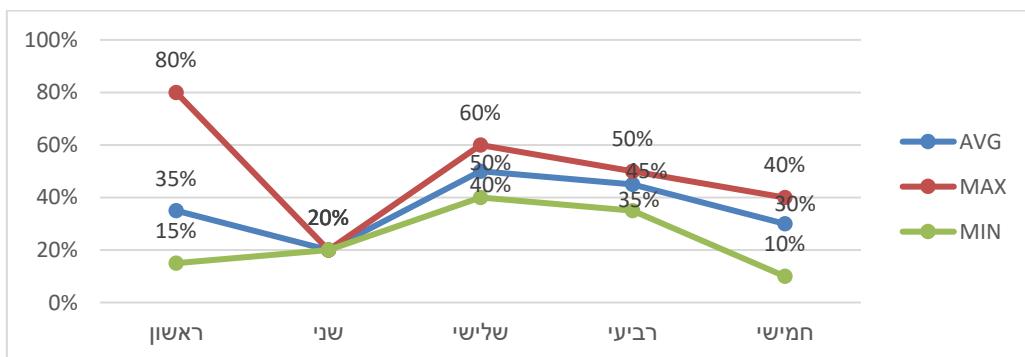


כדי להציג את אחוזי תפוזת המחשבים עבור כל יום בשבוע אנו שומרים עבור כל יום בשבוע שבדקנו את המינימום והממוצע ברשימה של היום בשבוע שהמתאים לו. כאן למשל נשמר את אחוזי השימוש של 6/10, 3/6, 1/4 תחת יום ראשון, וכך הלאה.

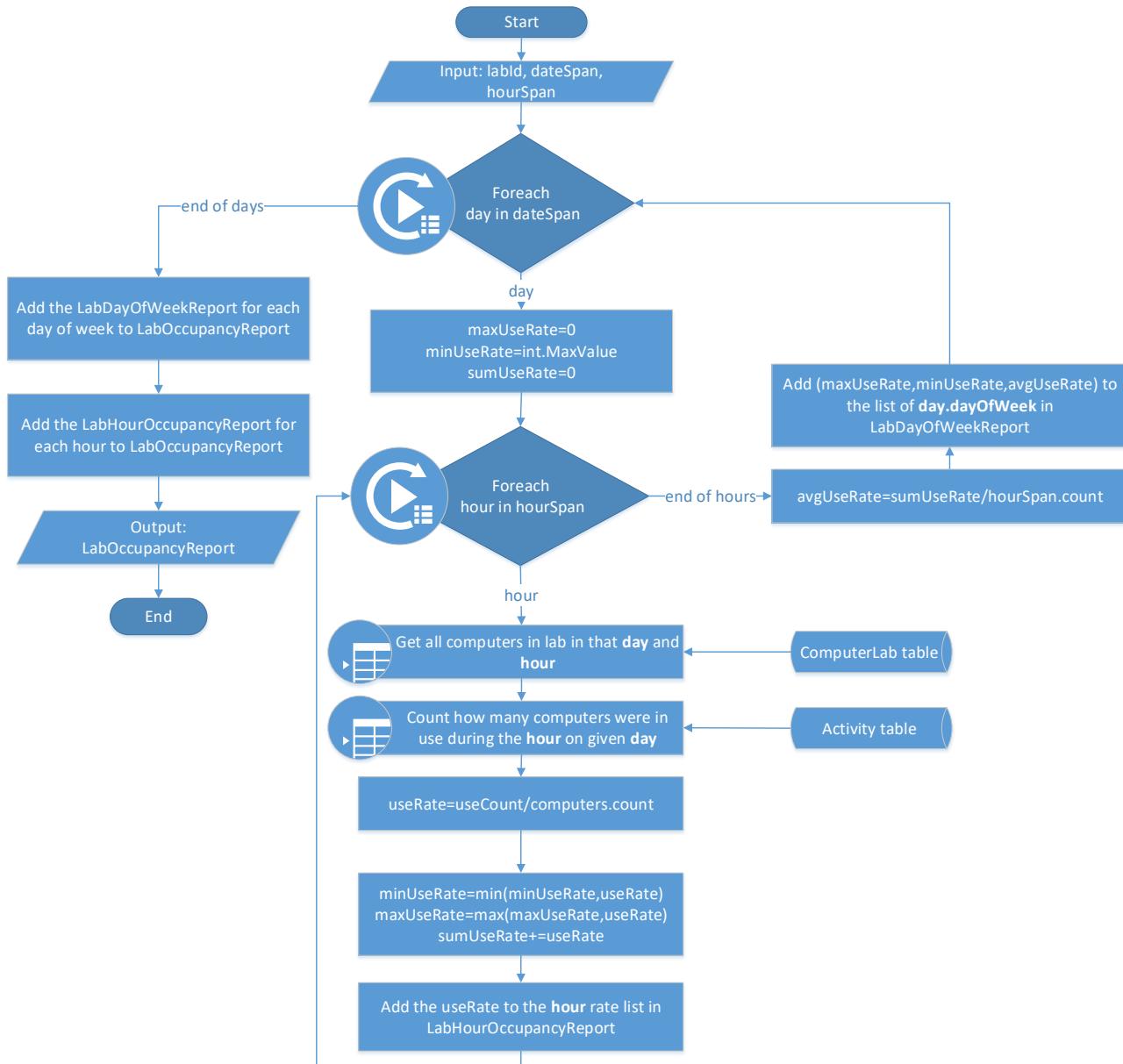
...	שלישי			שני			ראשון		
...	avg	min	max	avg	min	max	avg	min	max
...	50%	40%	60%	20%	20%	20%	25%	15%	40%
...	-	-	-	-	-	-	45%	20%	80%

עבור יום ראשון - אחוז השימוש המקסימלי הוא המקסימום מבין המקסימליים - 80%, המינימלי הוא המינימום מבין המינימליים 15%, והממוצע הוא ממוצע הממוצעים 35%.

דו"ח אחוז תפוזת המחשבים לפי ימים יראה כך :



ב. תיאור לוגי :



תרשים 10 - יצירת דוח אוחז תפוזת מחשבים

UC-18.3.6. תצוגת התראותא. **תיאור כללי:**

כל משתמש בעל הרשאה חלקית או מלאה יכול להיכנס לעמוד ההתראות ולראות את כל התتراות הקשורות למחשבים הנמצאים במחלקות אליוין יש לו הרשות גישה.

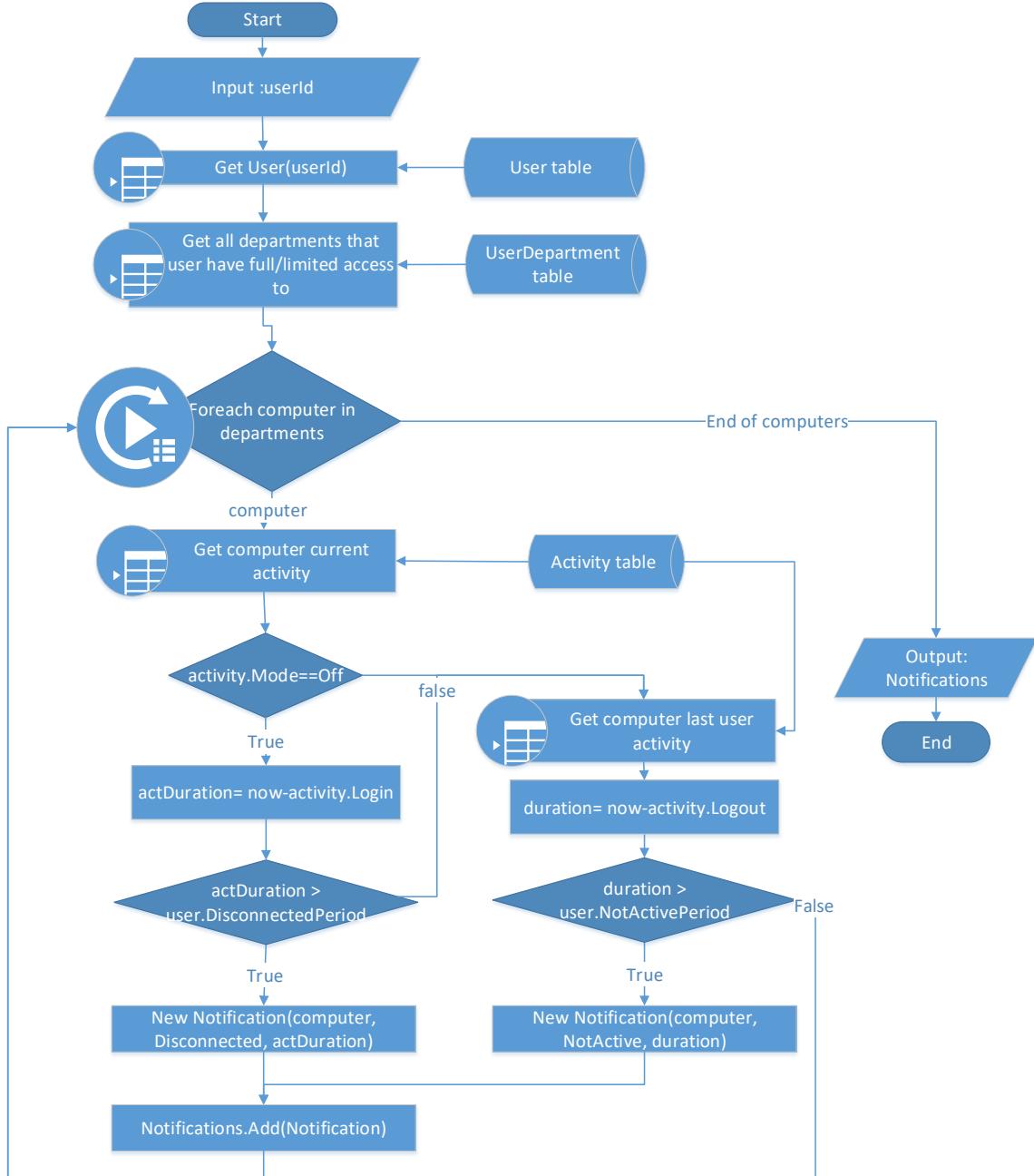
ישנן שתי סוגי התראות:

- 1) התראה אודוות מחשב שלא הטעינה אליו התחברות משתמשת תקופה ארוכה (מספר הימים של חוסר פעילות בעקבותיהם יש להציג התראה למשתמש מוגדרת ע"י המשתמש).

- (2) הטראה אודות מחשב שלא מגיב תקופה ארוכה (מספר הימים של חוסר תגובה בעקבותיהם יש להציג הטראה למשתמש מוגדרת ע"י המשתמש).

לכל מחשב תוצג סוג ה特朗אה שלו ומספר הימים שהוא לא בשימוש/אינו מגיב.

ב. תיאור לוגי:



תרשים 11 - תצוגת ה特朗אות

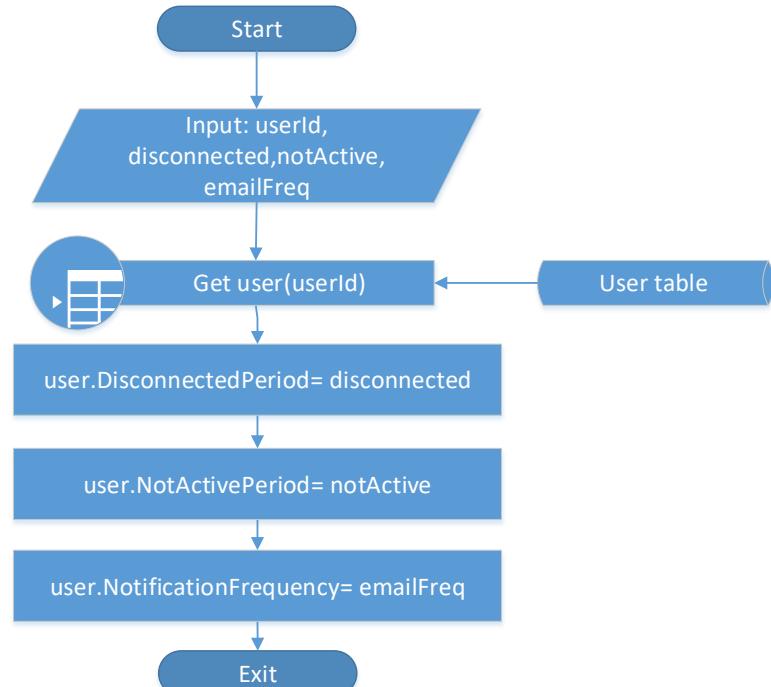
18.3.7. ניהול מדיניות ה特朗אות-9 UC

א. תיאור כללי:

כל משתמש בעל הרשאה גישה מלאה או חלקית יכול לשנות את הגדרת מדיניות התראות שלו. להגדר לאחר מכן ימם של חוסר פעילות של מחשב ירצה לראות התראה לגבי מחשב זה ולאחר כמה ימים של חוסר שימוש במחשב ירצה לראות על כך התראה.

בנוסף יש באפשרות המשתמש להגדיר לקבל מייל עם התראות על המחשבים, ניתן להגדיר את תזרות המיללים האלה: יומי/שבועי/חודשי או כלל לא.

ב. תיאור לוגי:



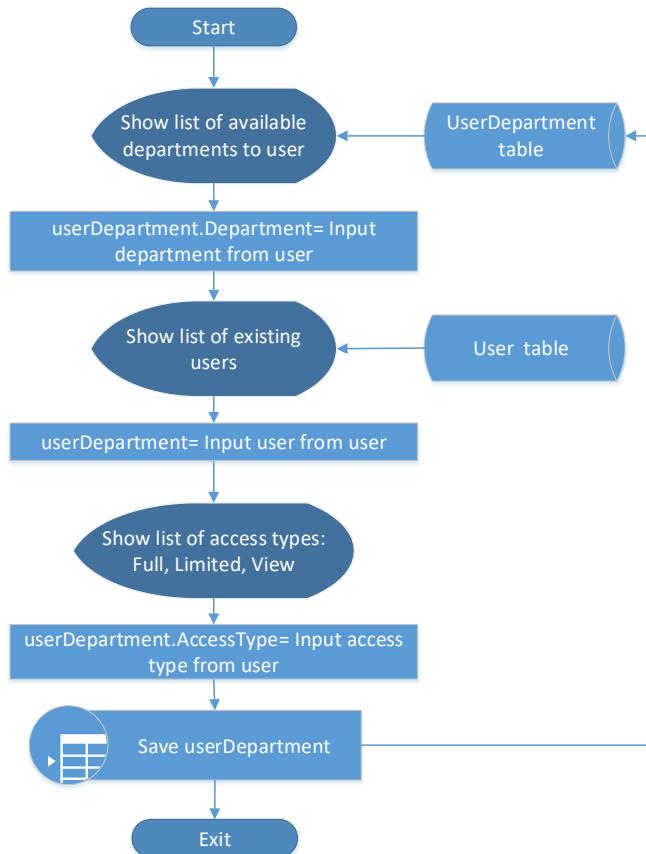
תרשים 12 – ניהול מדיניות התראות

18.3.8. הוספת הרשאות UC-10

א. תיאור כללי:

משתמש בעל הרשאה יכול להכנס לעמוד ההרשאות ולראות את כל ההרשאות הגישה למחלקות אליוין יש לו הרשאת גישה מלאה ובאפשרותו להוסיף הרשאות גישה למחלקות אלו. בהוספה הרשאת גישה הוא יבחר את המחלקה הרצויה מתוך המחלקות אליוין יש לו הרשאת גישה מלאה, יבחר את המשתמש לו ירצה לתת הרשאה מתוך רשימת המשתמשים ולבסוף יבחר את סוג הרשאת הגישה: מלאה/חלקית/כפיה.

ב. תיאור לוגי:



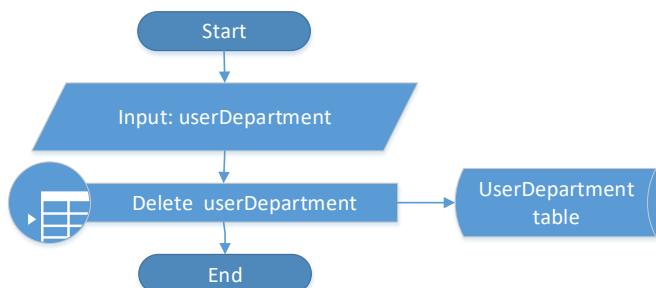
תרשים 13 - הוספת הרשות

UC-11.3.9. הורדת הרשות

א. תיאור כללי:

משתמש בעל הרשות יכול להכנס לעמוד הרשותות ולראות את כל הרשותות הגישה למחקות אליוין יש לו הרשות גישה מלאה ובאפשרותו למחוק הרשותות גישה ממחלקות אלו. בלחיצה על כפטור המבוקש על הרשותת המשתמש היא תימחק.

ב. תיאור לוגי:



תרשים 14 - הורדת הרשות

.18.4 אבטחת מידע

המערכת עלולה להכיל חומר רגיש אוודות סטודנטים כגון שמות משתמשים ושעות הפעולות שלהם במחשבים באוניברסיטה. כמו כן עלול להיות לפולני אינטראס לשבש את המידע על איסוף שעות פעילות המחשבים מסיבות Cálio או אחרות.

לצורך אבטחת המידע נקבעו נקודות בצדדים הבאים :

א. מתן הרשותות מינימאליות :

- 1) בכדי להתחבר לבסיס הנתונים הקמנו משתמש מיועד וננתנו לו הרשותות מינימאליות רק על הטבלאות שדרושות לכך שהמערכת תתפקד. בכך במידה של פריצה לא יכול הפורץ לגרום לנזקים משמעותיים בסיס הנתונים.
- 2) המערכת בשורת אוספת מידע על פעילות המחשבים דרך שרת Active Directory, לשם כך צרנו משתמש בעל הרשותות מינימאליות לקריאת בלבד הנתונים, ובעזרת ההרשאות שלו המערכת יכולה לאסוף את הנתונים הדרושים לה. במקרה של פריצה לא יכול המשתמש לעירוך שינויים ולא יוכל לשנות או למחוק דבר.

ב. הזדהות :

- 1) התחברות למערכת היא ע"י הזדהות מול שירות הזדהות קיימים של מערכות המחשב. משתמש יכול להתחבר למערכת רק לאחר שעבר הזדהות מול השירות הנ"ל. כניסה למערכת הניהולית תתאפשר רק למשתמשים אשר עברו את הזדהות מערך המחשב ובנוסף בדיקה אם המשתמש קיים בסיס הנתונים של המערכת.
- 2) למערכת היררכיית הרשותות, לכל משתמש מוגדר לאיזה מחלקות באוניברסיטה יש לו הראה ומהו סוג הרשאה, כך שרק משתמשים בעלי הרשותות מתאימות במערכת יכולים לבצע שינויים ורק בתוך המחלקות להן יש להם הרשאה.
- 3) בזמן הקשת הсистемה התווים חסומים ע"י כוכביות למניעת גניבת סיסמאות ע"י צופים.

ג. בידוד :

- 1) המערכת וכל חלקייה נמצאים בתחום הרשות הפרטיט של האוניברסיטה ולא פתוחים לרשות הציבורית. אין גישה לאתר, לבסיס נתונים ולשרותים מחוץ לתחומי האוניברסיטה (מלבד התחברות מאובטחת עם VPN של האוניברסיטה).
- 2) הספריות בהן משתמש המערכת ממוקמות בשרת, באופן זה לא נשלחות בקשנות מתוך המערכת לאתרם מחוץ לרשות של האוניברסיטה. כמו כן השירותים אינם מחוברים לרשות ציבורית או לאינטרנט.
- 3) בשרת IIS האתר רץ מתיקיה שנמצאת בគן D ולא בគן C שבו נמצאים קבצי המערכת וקבצים חשובים אחרים. בנוסף כל התיקיות והקבצים בתיקיה זו הם לקריאה בלבד, דבר המאפשר למערכת לעבוד אך במקרה של פריצה לא יכול הפורץ להגיע למידע חשוב ולא יוכל לבצע שינויים.

ד. טיפול בשגיאות :

- 1) הicken שיש צורך בהזנת נתונים של המשתמש מתבצעת בדיקה של הנתונים גם בצד הליכון בעזרת וליידציה ולאחר מכן מתבצעת בדיקה נוספת נספחת של הנתונים בשורת לפני הכנסתם לבסיס הנתונים.
 - 2) טיפול בחיריגות – עטפנו במערכת חלק קוד שעלולים ליצור שגיאות. שימוש בtry-catch מונע מהחיריגות שלא מטופלות לא לחושף פרטיים של המודול למשתמשים כמו למשל שמות של טבלאות ופונקציות.
 - 3) האטר לא מחזיר הודעה שגיאה מצד השירות ליישומי ליקוח – יש דף שגיאות מובנה שלא מראה את השגיאה שהראתה, כדי למנוע הבנה של מבנה המערכת מתוך השגיאות.
- ה. גישה לבסיס הנתונים :
- 1) חיבור מינימאלי לבסיס הנתונים. קשר עם בסיס הנתונים נפתח בתחילת חלק הקוד שדורש חיבור לבסיס נתונים ונסגר ישר בסוף החלק שדורש את הבסיס נתונים.
 - 2) הגישה למילוי הנתונים מתבצעת ע"י שימוש ב-LINQ (Language Integrated Query) – איננו מרכיבות באמצעות מניפולציה של מחרוזות או שרשור, והן אינן רגישות להתקפות הזורקת SQL מסורתיות.
 - 3) המערכת שולפת נתונים הכרחיים בלבד מהבסיס נתונים בכדי למנוע עומסים והעברה של מידע לא רלוונטי ברשות.
- ו. שרת המידע המכיל את המערכת נמצא פיזית במקום מאובטח בתוך האוניברסיטה.

18.5. בדיקות

בדיקות שנעשו במהלך הפיתוח :

- א. מעבר על הקוד הנוכחי
- ב. בדיקות ייחודית בעזרת Test Visual Studio

בענו בבדיקות ייחודית עבור פונקציונליות ייצור הדוחות וההתראות. הבדיקות ש她们 בפרויקט נפרדו אשר משתמשים במודולים של מערכת CAMS, ניתן להיכנס לפרויקט הבדיקות ולראות את כל הבדיקות הקיימות ולהרין בכל זמן נתון. כך אחר ביצוע כל שינוי הרצינו את הבדיקות מחדש ווידאו את תקינות הפונקציונליות.

אחד המודולים לו ביצעו בבדיקות ייחודיה היה מודול ייצור דוח אחזו – יצרנו דוחות שונים ובדקנו כי הפלט המתתקבל תקין ובדקנו מקרי קצה וביבים וקלטים שונים. למשל עבור ייצור דוח אחזו שימוש נבדקה ייצור הדוח עם ובלוי קלילת סופי שבוע, כאשר ישנה תזוזה של מחשבים בין כיתות המחשב בתאריכים עברים מבקשים להפיק את הדוח, כאשר יש חפיפות בין זמני השיעורים וזמן התחברות המשתמשים ועוד. להלן חלק מהבדיקות שעשינו עבור מודול זה :

מספר	מטרה	קלט	השימוש הצפוי פלט צפוי - אחוז	הצלחה/ כשלון
1	יצירת דוח אחוז שימוש עבור כיתת מחשב עם מחשב אחד.	דוח : 16/2/10 , 09:00-19:00 נתוני כיתה : מחשב יחיד במעבדה בתאריכים - 15/2/10- 15/3/10 . פעילות משתמש : 10:00-12:00 16/2/10	כולל שיעורים: 20% לא כולל: 20%	הצלחה
2	יצירת דוח אחוז שימוש עבור כיתת מחשב עם מחשב אחד, כאשר רק חלק פעילות המחשב נכנסת לטוח שעות הדוח.	דוח : 16/2/10 , 11:00-13:00 נתוני כיתה : מחשב יחיד במעבדה בתאריכים - 15/2/10- 15/3/10 . פעילות משתמש : 10:00-12:00 16/2/10	כולל שיעורים: 50% לא כולל: 50%	הצלחה
3	יצירת דוח אחוז שימוש עבור כיתת מחשב עם מחשב אחד, כאשר פעילות המחשב לא נכנסת בטוח שעות בדוח.	דוח : 16/2/10 , 13:00-15:00 נתוני כיתה : מחשב יחיד במעבדה בתאריכים - 15/2/10- 15/3/10 . פעילות משתמש : 10:00-12:00 16/2/10	כולל שיעורים: 0% לא כולל: 0%	הצלחה
4	יצירת דוח אחוז שימוש עבור כיתת מחשב עם מחשב אחד בעל שני פעילויות, כאשר פעילויות המחשב לא נכנסות בטוח שעות בדוח.	דוח : 16/2/10 , 12:00-13:00 נתוני כיתה : מחשב יחיד במעבדה בתאריכים - 15/2/10- 15/3/10 . פעילות משתמש : ב 16/2/10 13:00-15:00 , ב 10:00-12:00 13:00-15:00 , 10:00-12:00	כולל שיעורים: 0% לא כולל: 0%	הצלחה
5	יצירת דוח אחוז שימוש עבור כיתת מחשב עם מחשב אחד בעל שתי פעילויות, כאשר שתי פעילות המחשב נכנסות لطוח שעות הדוח.	דוח : 16/2/10 , 08:00-18:00 נתוני כיתה : מחשב יחיד במעבדה בתאריכים - 15/2/10- 15/3/10 . פעילות משתמש : ב 16/2/10 13:00-15:00 , ב 10:00-12:00 13:00-15:00 , 10:00-12:00 13:00-15:00	כולל שיעורים: 40% לא כולל: 40%	הצלחה
6	יצירת דוח אחוז שימוש עבור כיתת מחשב עם מחשב אחד בעל שתי פעילויות, כאשר חלקים משתי פעילויות המחשב נכנסות לטוח שעות הdoch.	דוח : 16/2/10 , 11:00-14:00 נתוני כיתה : מחשב יחיד במעבדה בתאריכים - 15/2/10- 15/3/10 . פעילות משתמש : 13:00-15:00 , 10:00-12:00 16/2/10 13:00-15:00	כולל שיעורים: 66.66% לא כולל: 66.66%	הצלחה
7	יצירת דוח אחוז שימוש של יומיים עבור כיתת מחשב עם מחשב אחד שאין לו פעילויות ביום הdoch.	דוח : 17/2/10-18/2/10 , 08:00- 18:00 נתוני כיתה : מחשב יחיד במעבדה בתאריכים - 15/2/10- 15/3/10 . פעילות משתמש :	כולל שיעורים: 0% לא כולל: 0%	הצלחה

		ב10:00- ,10:00-12:00 16/2/10 15:00		
8	הצלחה	כולל שיעורים: 0% לא כולל: 0%	דוח : 08:00-12:00 ,15/3/10 נתוני כיתה : מחשב יחיד במעבדה בתאריכים -15/2/10 עד 15/3/10 משתמש : ב10:00- 16/2/10 עד 10:00- 15/3/10 . פעילות 13:00-15:00 ,12:00	יצירת דוח אחוז שימוש עבור כיתת מחובע עם מחשב אחד שיצא מכיתת המחשב במהלך הדוח. קיימות למחשב פעילות בתווך הדוח אך הוא לא נמצא בכיתת המחשב בזמן זה.
9	הצלחה	כולל שיעורים: 15.38% לא כולל: 15.38%	דוח : 08:00-18:00 ,9/10/10-16/10/10 נתוני כיתה : מחשב יחיד במעבדה בתאריכים 15/10/10-(10:00-10/10/10) משתמש : 10:00-19:00 10/10/10 (12:00)	יצירת דוח אחוז שימוש עבור כיתת מחובע עם מחשב אחד שיצא מכיתת המחשב במהלך הדוח. קיימות למחשב פעילות בתווך הדוח כאשר הוא בכיתת המחשב.
10	הצלחה	כולל שיעורים: 20% לא כולל: 20%	דוח : 09:00-19:00 ,16/2/10 נתוני כיתה : מחשב יחיד במעבדה בתאריכים -15/2/10-15/3/10 משתמש : 10:00-12:00 16/2/10 . פעילות 10:00-12:00 16/2/10 שיעור :	יצירת דוח אחוז שימוש עבור כיתת מחובע עם מחשב אחד. חפיפה מלאה בין פעילות משתמש לפעילויות שיעור.
11	הצלחה	כולל שיעורים: 30% לא כולל: 20%	דוח : 09:00-19:00 ,16/2/10 נתוני כיתה : מחשב יחיד במעבדה בתאריכים -15/2/10-15/3/10 משתמש : 10:00-19:00 16/2/10 . פעילות 09:00-11:00 16/2/10 שיעור :	יצירת דוח אחוז שימוש עבור כיתת מחובע עם מחשב אחד. חפיפה חלקית בין פעילות משתמש לפעילויות שיעור.
12	הצלחה	כולל שיעורים: 50% לא כולל: 20%	דוח : 09:00-19:00 ,16/2/10 נתוני כיתה : מחשב יחיד במעבדה בתאריכים -15/2/10-15/3/10 משתמש מוכל בפעילויות שיעור :	יצירת דוח אחוז שימוש עבור כיתת מחובע עם מחשב אחד. פעילות משתמש מוכל בפעילויות שיעור.
13	הצלחה	כולל שיעורים: 40% לא כולל: 20%	דוח : 09:00-19:00 ,16/2/10 נתוני כיתה : מחשב יחיד במעבדה בתאריכים -15/2/10-15/3/10 משתמש : 10:00-19:00 16/2/10 . פעילות 13:00-15:00 16/2/10 שיעור :	יצירת דוח אחוז שימוש עבור כיתת מחובע עם מחשב אחד. אין חפיפה בין פעילות.

14	הצלחה	כולל 75%: לא כולל: 50%	דוח : 11:00-13:00 , 16/2/10 נתוני כיתה : מחשב יחיד במעבדה בתאריכים - 10/15/2/10 15/3/10 . פעילות משתמש : שיעור : 10:00-19:00 16/2/10 שיעור : 13:00-15:00 16/2/10 10:30-12:30	יצירת דוח אחז שימוש עבור כיתת מחשב עם מחשב אחד. פעילות משתמש ושתי פעילותות מחשב. חפיפה חלקית בין פעילותות.
15	הצלחה	כולל 100%: לא כולל: 0%	דוח : 13:00-15:00 , 16/2/10 נתוני כיתה : מחשב יחיד במעבדה בתאריכים - 10/15/2/10 15/3/10 . פעילות משתמש : שיעור : 10:00-19:00 16/2/10 שיעור : 13:00-15:00 16/2/10 10:30-12:30	יצירת דוח אחז שימוש עבור כיתת מחשב עם מחשב אחד. רק פעילות שיעור בטוח זמני הדוח.

טבלה 13 - בדיקות ייחוד

18.6 ניסויים

18.6.1 ניסוי בדיקת תהליכי הפקת דוחות

- א. מיקום הניסוי: אוניברסיטת בן גוריון
- ב. הכנה לניסוי: השתמשנו בנתונים אותם אספה המערכת הקיימת ביום על פעילותות המחשבים. מערכת זו שומרת עבור כל מחשב קובץ שמכיל את המידע על כל הפעולות שלו. לקחנו את הקבצים של תיעוד פעילותות המחשבים עבור אחת המעבדות והכנו סקירהפט שマפְרָסֶר את הפעולות, מתאים אותו לפורמט של בסיס הנתונים שלנו ומכניס אותו אליו.
- ג. מהלך הניסוי: הפקנו דוחות בטוחי תאריכים שונים, טוחי שעות שונים, עם ובלי התחשבות בסופי שבוע.
- ד. תוצאות ומסקנות הניסוי: רأינו כי אכן המערכת מתמודדת עם הפקת דוחות עבור פרמטרים שונים.

18.6.2 ניסוי בדיקת תקינות איסוף הנתונים ותצוגת תוצאה נוכחית

- א. מיקום הניסוי: אוניברסיטת בן גוריון
- ב. הכנה לניסוי: יצרנו בתוך המערכת את כיתות המחשב הציבוריות השיווקות למחלקה להנדסת מערכות תוכנה ומדע, הוספנו לכל כיתה את המחשבים שישיכים לה במציאות וסידרנו אותם על המפה כפי שהם מסודרים בפועל.
- ג. מהלך הניסוי: ביצענו הריצה של המערכת בסביבת האוניברסיטה במשך 10 שעות, בזמן זה המערכת אספה את פעילותות המחשבים של כיתות המחשב של המחלקה להנדסת מערכות תוכנה ומדע, תוך כדי הריצה הדלקנו וכיבינו מחשבים, ביצענו התחברויות וניתוקים של משתמשים שונים כדי לוודא שאכן המידע מתעדכן במסיס הנתונים ומוצג למשתמש באופן תקין
- ד. תוצאות ומסקנות הניסוי: רأינו כי אכן המערכת מתפרקת באופן הרצוי, אוסף את הנתונים באופן תקין ומציגה אותם כראוי.

18.7 התואששות מתקלות .

- א. הגדרנו את בסיס הנתונים כך שהוא מבצע גיבויים אוטומטיים פעם ביום.
- ב. בשרת SQL 2017 ישנו קובץ שימושי בשם `TRNZEKCIOT`. קובץ זה עוזר לשרת לבצע פעולות להთואששות מתקלות:
- (1) אם מנוע מסד הנתונים מזיהה שגיאה כגון אובדן תקשורת עם לוקח, רשותם היום משמשות לבצע חזרה לאחר (rollback) של השינויים שבוצעו בטרנזקציה לא שלמה.
 - (2) במידה והשרת נכשל, כאשר הוא מופעל מחדש, הוא מפעיל שחזור של כל מסד הנתונים. כל עסקה לא שלמה שנמצאת ביום הטרנזקציות מוחזרת לאחר מכן כדי לוודא שהשלמות של מסד הנתונים נשמרת.
- ג. כאשר בקוד יש טנזקציות מורכבות מכמה שמיראות של נתונים למסד הנתונים הן מבוצעות תחת סקופ של טנזקציה ורק בסיום מתבצע `commit`. אם מסיבה מסוימת יש שגיאה באמצעות rollback על הפעולות שבתוך הסקופ הניל.
- ד. במידה ושרת האינטרנט נופל, כאשר הוא יופעל מחדש המערכת מופעלת מחדש ואתה פונקציה `Startup` מתחילה זו קראנו לכל התהליכיים שרוצים ללא הפסקה כגון איסוף הנתונים, שילוח התוצאות, איסוף שיעורים מתוכננים ועוד.

18.8 מבנה התקיינות והקבצים .

הערה – חלק מעמודי האינטרנט המפורטים מטה מקבלים פרמטרים בשיטת GET או POST ומשנים את התנהוגות בהתאם. המידע הניל הוא תיאור כללי בלבד.

- א. תיكيית View – מכילה את דפי התצוגה מוחולקים לפי נושאים.

שם תיKİיה	שם קובץ	הסבר
Account	Login.cshtml	מכיל את דף ההתחברות למערכת ניהול
Buildings	Index.cshtml	מציג רשימה של כל הבניינים
	Edit.cshtml	דף עריכה של שם בניין
	Delete.cshtml	דף מחיקה של בניין
Departments	Index.cshtml	מציג רשימה של כל המחלקות
	Create.cshtml	יצירה של מחלקה חדשה
	Edit.cshtml	דף עריכה של מחלקה
	Delete.cshtml	דף מחיקה של מחלקה

דף ראשי של המערכת בה יש טבלה של CiTiotot המחשב	Index.cshtml	
יצירה של CiTiotot מחשב חדשה	Create.cshtml	Labs
דף Edite של CiTiotot מחשב	Edit.cshtml	
דף Delete של CiTiotot מחשב	Delete.cshtml	
מציג את ההתראות של המשתמש	Index.cshtml	Notifications
הגדירה מתי לקבל מיילים ועל מה לקבל התראות	Edit.cshtml	
דף יצירה של דוח	Create.cshtml	
תצוגה של דוח ממוצע שעות פעילות	AverageUsage.cshtml	
תצוגה של דוח פעילות מחשבים לפי שעות	LabOccupancy ReportHour.cshtml	Reports
תצוגה של דוח פעילות מחשבים לפי ימים	LabOccupancy ReportDay.cshtml	
תסdìר המערכת, מכיל כוורת עליונה ותחתונה לכל דפי המערכת.	_CAMSLayout.cshtml	Shared
מציג רשימה של משתמשים, מחלקות וההרשאות שיש לכל משתמש למחלקה	Index.cshtml	
יצירת הרשאה חדשה	Create.cshtml	Users
דף Edite של הרשאה	Edit.cshtml	
דף Delete של הרשאה	Delete.cshtml	
מגדר למערכת איזה דף להציג בהתחלה	_ViewStart.cshtml	

טבלה 14 - מבנה תיקייה View

ב. **תיקיות Scripts – מכילה את דפי JavaScript שהמערכת משתמש בהן דפים עיקריים :**

שם תיקייה	שם קובץ	הסביר
CAMS_scripts	Labs.js	fonktsionality של CiTiotot המחשב

פונקציונליות של דפי התצוגה של התראות	Notifications.js	
פונקציונליות של דפי התצוגה של דוחות	Reports.js	
פונקציונליות של דפי התצוגה של הרשות	Users.js	
מכיל פונקציונליות של גיררת המחשבים בדף העריכה של כיתות המחשב	jquery.ui.multidraggable-1.8.8.js, ui.multidraggable.js	multidraggable
עיצוב ובניית גרפים לדוחות	Chart.js	
פונקציונליות של בחירת תאריכים בדף יצירת הדוחות	datepicker-he.js	
פונקציונליות של בחירת שעות בדף יצירת הדוחות	jquery-TimePicker-1.0.0.js	

טבלה 15 - מבנה תייקית Scripts

ג. תייקית Models – עיקר הלוגיקה והאלגוריתמים

שם קובץ	הסבר
AccessViewModel.cs	מחלקה עבור תצוגת הרשאת גישה של משתמש למחלקה
AccountViewModels.cs	מחלקה עבור חיבור של משתמש למערכת
ActivitiesModel.cs	עבודה וניהוות עם פעילויות המחשבים
CAMS_DatabaseModel.edmx	המודל של מסד הנתונים
CAMS_DatabaseModel.edmx.sql	סקרייפט של מבנה מסד הנתונים
ComputerReport.cs	מחלקה עבור דוח אחוז שימוש ממוצע עבור מחשב
LabOccupancyReport.cs	מחלקה עבור דוח אחוז תפוסת מחשבים בכיתת מחשב
LabReport.cs	מחלקה עבור דוח אחוז שימוש ממוצע עבור כיתת מחשב

מחלקה עבור תצוגת דוח	LabsReportViewModel.cs
מחלקה עבור תצוגת כיתת מחשב	LabsViewModel.cs
מחלקה עבור תצוגת התראות	NotificationViewModel.cs
מודל ליצירת דוחות	ReportModel.cs

טבלה 16 - מבנה תקנית Models

ד. תקנית Images – מכילה תמונות שמשמשות את האתר

שם קובץ	הסביר
black.png	תמונה של מחשב לא מחובר
clear.png	תמונה של מחשב בעריכה
green.png	תמונה של מחשב פנוי
red.png	תמונה של מחשב תפוס
logoHome.png	תמונה שנמצאת בគורתה של האתר

טבלה 17 - מבנה תקנית Images

ה. תקנית Controllers – מכילה את הקונטROLLERים האחראים לעובדה מול בסיס הנתונים

שם קובץ	הסביר
AccountController.cs	פונקציונליות של התחברות למערכת
ActivitiesController.cs	פונקציונליות של הפעולות בסיסיות וככליות ממנה ירשים
BaseController.cs	קונטROLLER בעל פונקציונליות בסיסיות וככליות ממנה ירשים יתר הקונטROLLERים.
BuildingsController.cs	פונקציונליות של הבניינים בסיסיים.
ComputersController.cs	פונקציונליות של המחשבים בסיסיים.
DepartmentsController.cs	פונקציונליות של המחלקות בסיסיים.
LabsController.cs	פונקציונליות של כיתות המחשב בסיסיים.
NotificationsController.cs	פונקציונליות של התראות בסיסיים.
ReportsController.cs	פונקציונליות של הדוחות בסיסיים.

פונקציונליות של המשתמשים בסיס הנתונים.	UsersController.cs
--	--------------------

טבלה 18 ~ מבנה תייקית Controllers

- ו. תייקית Content מכילה את קבצי העיצוב CSS של האתר.
- ז. תייקית App_Start מכילה קבצי קונפיגורציה של האתר.

19. תצוגת המערכת

19.1. דף התחברות למערכת



התחברות

שם משתמש

סיסמה

זכור אותי

© 2018 כל הזכויות שמורות המחלקה להנדסת מערכות תוכנה ומדע – אוניברסיטת בן-גוריון בנגב

דף ראשי לסטודנטים .19.2



כיתות מחשוב

חפש לפי שם מחלקה / בניין:

	מספר עדות פניוות	שיעור מתחננים להיום	כיתה		בנין	מחלקה
תפוצה נוכחית	10	אין	629	72 - בניית אקור, גולדשטיין-גורו למדעי הרוח והחברה		אנתרופולוגיה
תפוצה נוכחית	8	אין	249	72 - בניית אקור, גולדשטיין-גורו למדעי הרוח והחברה		גיאוגרפיה
תפוצה נוכחית	5	אין	122			גיאולוגיה
תפוצה נוכחית	13	אין	028	63 - בניית האורי ואובי שרמן למחלקה להנדסה גרעינית		הנדסה ביורופאית
תפוצה נוכחית	9	11:00-12:00, 10:00-11:00	237	59 - בניית הווולד 'ה' פוטשור להנדסה כימית ולהנדסת חומרים		הנדסה כימית
תפוצה נוכחית	2	אין	650			הנדסה בניין
תפוצה נוכחית	17	13:00-15:00	135	59 - בניית הווולד 'ה' פוטשור להנדסה כימית ולהנדסת חומרים		הנדסת חומרים
תפוצה נוכחית	6	אין	118	33 - בניית זלוטובסקי למחלקה להנדסת חשמל ומחשבים		הנדסת חשמל

עמוד 1 מתוך 7

1 2 3 4 5 6 7 <<

© 2018 כל הזכויות שמורות למחלקה להנדסה מערכות תוכנה ומידע - אוניברסיטת בן-גוריון בנגב

דף צפיה בכיתת מחשב 19.3



דוחות | ביצועים חשובים | התראות | הרשות

Page 1 of 1

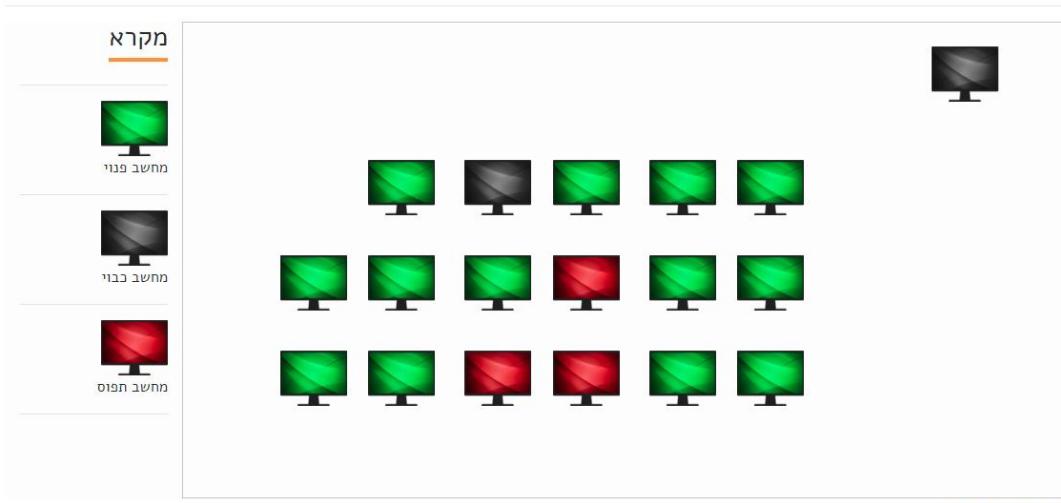
מפעלים וורכתיים

טפומת גורמים

כפיון: 96 – גניון קROL ומרקוס ויינשטיין לבקשת מטרות מידע ואבטחת סייבר

103 / 2017

-107 (iii)



ג'ורו | חזרה לרשימה

.19.4 דף>Edit ביתת מחשב
[דוחות](#) | [כיתות מחשב](#) | [התראות](#) | [הרשאות](#)
עריבה

בנין:

96 – בניין קרול ומרקס ויינשטיין להנדסת מערכות מידע ובטחת סייבר

כיתה:

-107

LB-107-14	LB-107-11	LB-107-8	LB-107-5	LB-107-2	
LB-107-17	LB-107-15	LB-107-12	LB-107-9	LB-107-6	LB-107-3
LB-107-18	LB-107-16	LB-107-13	LB-107-10	LB-107-7	LB-107-4

מחשבים

TS12	
CAMPUS14,R2	
CAMPUSP12	
CASA	
CASA2	
BGU-WDEV1	
DEMO-VISION	
DONOR-MT-TST	
DONOR-MT	
DRPARISTOA4	
HL-SQL	
CHEMENGDC1	

גודל מחשבים: גדול בינוני קטן
מחק מחשבים נבחרים
שמור

ביטול

.19.5 דף הוספה ביתת מחשב
[דוחות](#) | [כיתות מחשב](#) | [התראות](#) | [הרשאות](#)
ביתת מחשב חדשה

מחלקה

בניין

בניין חדש

כיתה

צור

זירה לרשימה

.19.6 דוח הפקת דוח

דוחות | בתיות מחשב | התראות | הרשות

דוחות**דוח חדש**

סוג דוח
מוצע שעות פעילות
תאריךים
מ - 17/06/2018 עד 03/06/2018 לא כולל סופי שבוע
שעות
מ - 21:00 עד 09:00 כל היום
כתובות מחשב
בחר הכל / הספר הכל
הנדסת מערכות תוכנה ומידע
96 - בניית קROL ומרקום ויינשטיין להנדסת מערכות מידע ובאטחתת סייבר ביתה -107 96 - בניית קROL ומרקום ויינשטיין להנדסת מערכות מידע ובאטחתת סייבר ביתה 3003 96 - בניית קROL ומרקום ויינשטיין להנדסת מערכות מידע ובאטחתת סייבר ביתה -105
הפקולטה לניהול
צור

.19.7 דוח ממוצע שעות פעילות

דוחות | בתיות מחשב | התראות | הרשות

דוח פעילות מחשבים

מתאריך 03/06/2018 עד - 17/06/2018 כולל סופי שבוע. בין השעות 09:00 עד - 21:00

מחלקה	בניין	כיתה	מספר עמדות	אחסן שימוש	אחסן שימוש מוצע בול שיבוץ חדרים
הנדסת מערכות תוכנה ומידע	96 - בניית קROL ומרקום ויינשטיין להנדסת מערכות מידע ובאטחתת סייבר	-107	18	22.48	22.48
הנדסת מערכות תוכנה ומידע	96 - בניית קROL ומרקום ויינשטיין להנדסת מערכות מידע ובאטחתת סייבר	-105	31	54.48	54.48
הנדסת מערכות תוכנה ומידע	96 - בניית קROL ומרקום ויינשטיין להנדסת מערכות מידע ובאטחתת סייבר	003	31	61.02	61.02

צור דוח חדש

דוח פעילות מחשבים לפי שעות .19.8

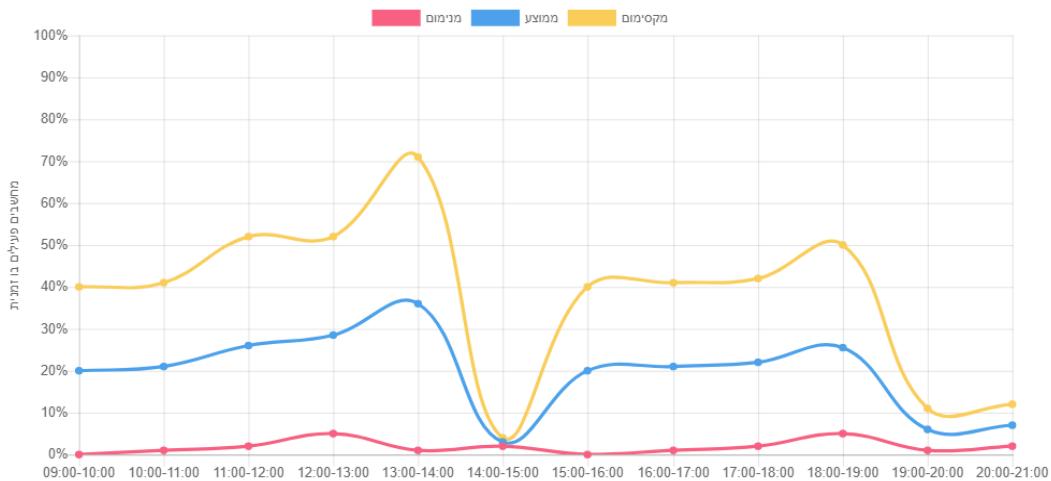
אחוז תפוקת מחשבים בכיתות לפי שעות

מתאריך 15/04/2018 עד - 19/04/2018 כולם סופי שבוע, בין השעות 09:00 עד - 21:00

פירוט ביתה

בניין: 96 - בניין קROL ומרקום ויינשטיין להנדסת מערכות מידע ובטחת סייבר

כיתה: 707



דוח התראות ועריכת התראות .19.9



אוניברסיטת בן-גוריון בנגב
Ben-Gurion University of the Negev

דוחות | בתיות מחשב | התראות | הרשות

התראות

ערוך הגדרות התראות

חפש לפי שם מחלקה / בניין:

מחלקה	בניין	התראות		
		שם עמדת	שם עמדת	פירוט התראה
אפקט בכתה	-107	לא מחובר 6 ימים	lb-107-1	לא מחובר 6 ימים
		לא בשימוש 28 ימים	lb-107-2	לא בשימוש 28 ימים
		לא בשימוש 27 ימים	lb-107-3	לא בשימוש 27 ימים
		לא בשימוש 8 ימים	lb-107-4	לא בשימוש 8 ימים
		לא בשימוש 15 ימים	lb-107-5	לא בשימוש 15 ימים
		לא בשימוש 2 ימים	lb-107-6	לא בשימוש 2 ימים
		לא בשימוש 10 ימים	lb-107-7	לא בשימוש 10 ימים
		לא בשימוש 2 ימים	lb-107-8	לא בשימוש 2 ימים
		לא בשימוש 2 ימים	lb-107-9	לא בשימוש 2 ימים
		לא בשימוש 10 ימים	lb-107-10	לא בשימוש 10 ימים
		לא משמש 64 ימים	lb-107-11	לא משמש 64 ימים
		לא בשימוש 28 ימים	lb-107-12	לא בשימוש 28 ימים
		לא בשימוש 55 ימים	lb-107-13	לא בשימוש 55 ימים
		לא בשימוש 28 ימים	lb-107-14	לא בשימוש 28 ימים
		לא בשימוש 63 ימים	lb-107-15	לא בשימוש 63 ימים

עריבת העדפת התראות

האם לשלווח התראות למייל?

 כן, אחת לחודש

הגדר לאחר כמה ימים להתראה עברו:

מחשב ללא חיבורת

 4

מחשב לא משמש מחובר

 6
שמור**חזרה להתראות**

© 2018 כל הזכויות שמורות למחלקה להנדסת מערכות תוכנה ומדע – אוניברסיטת בן-גוריון בנגב

דף הרשות, עריכת הרשות והוספה לרשותה 19.10.**הרשות**

הוסף הרשות חדשה

סוג הרשות	מחלקה	כתובת
ערוך מחק	כלאה	הפקולטה לניהול nasirov@post.bgu.ac.il
ערוך מחק	חלקית	הנדסת מערכות תוכנה ומדע
ערוך מחק	צפיה בלבד	הפקולטה לניהול cams@post.bgu.ac.il

© 2018 כל הזכויות שמורות למחלקה להנדסת מערכות תוכנה ומדע – אוניברסיטת בן-גוריון בנגב

ערוך הרשאה

דו"ל

adiron@post.bgu.ac.il

מחלקה

הנדסת מערכות תוכנה ו מידע

סוג הרשאה

הרשאה מלאה

שמור

[חזרה להרשאות](#)הרשאה חדשה

דו"ל

adiron@post.bgu.ac.il

מחלקה

הפקולטה לניהול

סוג הרשאה

הרשאה מלאה

צור

[חזרה להרשאות](#)**.20. הטענת המערכת**

בתחילת תחילת הפיתוח עבדנו בסביבת פיתוח עם שרת SQL של Azure ועם שרת IIS express שהותקן מקומית במחשבים הפרטיים שלנו. לאחר פיתוח חלק ניכר מהמערכת הבאנו אותה לתצוגה ללקוח.

.20.1 הקמת תשתיית

לאחר הצגת המערכת לקוחות, הלקוח הקים לנו סביבת עבודה לבדיקות בתוך רשת האוניברסיטה והיא מדמה את סביבת הייצור.

הוקמו למען המערכת 4 שרתים :

- forest schema master : זהו ה Root domain (1)
 של Active Directory
 root domain child domain : שרת זה הוא public class domain (2)
 דומיין אוטנטיקציה (3)
 Windows Server 2016 member server : שרת חבר windows 10 גרסה 10. (4)

בנוסף הוקמו שני מחשבים וירטואליים לבדיקות אחד מבוסס windows 10 והשני windows 10. על שרת Windows Server 2017 התקנו SQL Server 2017 ו IIS.

.Entity Framework Code first הקמנו את בסיס נתונים ע"י

הגדכנו משתמשים בעלי הרשות מינימלית לפקוד המערכת בשרת SQL ובשרת אוטנטיקציה.

20.2. תחזוקה

ככל, המערכת אינה זקופה לתחזוקה שוטפת ולbianco שינויים ע"י אנשי IT. למרות זאת, יתכו מקרים בהם תידרש התערבות של אנשי IT :

- (1) שינויים בהגדרת אפליקציה (קובץ config web.config)
- (2) שינוי כתובות שרתים – IIS, SQL
- (3) גדילה משמעותית של בסיס הנתונים, איטיות בעבודה – ישנו צורך לנוקות רשותות ישנות של פעילויות ומחשבים מהמערכת על מנת לוrozen את פעולהمنع database.
- (4) שחזור של בסיס הנתונים.

21. אתגרים

לאורך הפרויקט התמודדנו עם אתגרים שונים ומגוונים :

- (1) הרשותות- על מנת למש את כל הפונקציונליות הנדרשת מהמערכת לקבל הרשותות מתאימות :
- א. איסוף הנתונים- איסוף הנתונים על מחשבים יכול להתבצע כאשר יש למערכת הרשות גישה לדומיינים להם שייכים המחשבים. בתור סטודנטיות מין המניין בעת התחברות למחשב באוניברסיטה יכולנו לגשת ולראות דרך כל PowerShell (בו משתמש המערכת שלנו לצורף איסוף הנתונים) את מצב המחשבים באותו הדומיין אך לא לדומיינים אחרים.
- ב. שילוף מחשבים קיימים בדומיין- בעת עריכה של כיתה, מוצגים למשתמש כל המחשבים אותם הוא יכול להוסר לкласс, מחשבים אלה נאספים תוך שימוש Active Directory, בעזרתו המערכת שולפת את כל המחשבים שייכים לדומיין. בדומה לאיסוף הנתונים כדי לגשת מידע זהה צריך הרשאה לדומיין.

המחסור בהרשאות המתאימות מאוד הגביל את היכולות של המערכת ולא יכולנו לבצע בדיקות מקיפות על המערכת כפי שהיינו רוצות.

(2) לצפות את כל התרחישים האפשריים ומרקם הקצה שלהם -

החל משלב מאוד מוקדם של מחזור חיiproject הינו צרכות להבין לעומק את מבנה הארגון (איך הרשות בניה, היררכיית ההרשאות, תהליכי אפשריים) כדי להתאים את המערכת בצורה הטובה ביותר ליותר לצרכים שלה. ניתוח הדרישות והבנת כל התרחישים האפשריים היה תהליך שחייב חקירה ומחשבה רבה ובהתאם לתהליך זה עיצבנו את המערכת.

(3) עבודה צוות-

עבודה בצוות וחלוקת העבודה לא תמיד הייתה פשוטה, לעיתים יש תלות בין חלקיו הפרויקט שמצריכה תיאום ותקשורת שוטפת.

(4) ניהול זמנים-

על מנת לעמוד בכל יעד הפרויקט הינו צרכות לקבועלוח זמנים, דבר שלא היה קל כיון שהקשה מאוד לצפות כמה זמן ייקח כל חלק (במיוחד בגלל שאין לנו ניסיון בניהול פרויקט בסדר גודל כזה) ויש לנקח בחשבון אירועים ותקלות בלתי מתוכננים שיכולים לקרות. אתלוח הזמנים הזה הינו צרכות להתאים לאורח החיים שלנו כסטודנטים-לקורסים שאנחנו לוקחות ולמשרות שלנו. התקשינו עם ניהול נכון של הזמן שלנו, התחלנו בשלב ההטמעה מאוחר יותר משציפינו והתמודדנו עם עומס רב בסוף הפרויקט.

(5) עבודה מול לקוחות-

הבנה של הדרישות הלkop והאופן שבו הוא רוצה שהמערכת תיראה. ביצענו ישיבות עם לקוחות בהן וידאנו שאנחנו מבינים אחד את השני כמו שצריך והמערכת עומדת בזמנים שלו.

(6) עיצוב המערכת בצורה פשוטה וברורה למשתמש -

היה מאוד חשוב לנו שהמערכת תהיה אטרקטיבית וaintuitive למשתמש, לצורך כך הושקעה מחשבה רבה בעיצוב ממשק המשמש.

22. סיכום

פיתחנו מערכת המציגת תפוצה נוכחית והיסטורית של מחשבים בכיתות המחשב הציבוריות באוניברסיטה בן גוריון. המערכת מאפשרת להפיק דוחות סטטיסטיים על פעילויות המחשבים.

דרישות המערכת הוגדרו על ידי מר מיכאל לובר, ראש מדור מחשב אישי במערך המחשב של האוניברסיטה. מערכת זו נותנת מענה לכל דרישות הלkop וצרכו כפי שאופיינו. בתכורתה המערכת נועה לשימוש, ברורה ומעוצבת. המערכת אוספת מרוחק את נתוני השימוש במחשבים הפזוריים באוניברסיטה.

המערכת תשמש בפועל את כל אחראי המחשב בפקולטות השונות, מנהליים ומקבלי החלטות בדרגות בכירות באוניברסיטה.

המערכת תשמש בפועל את מקבלי החלטות בהבנה עמוקה יותר על השימוש במחשבים ותעוזר בקבלת החלטות הקשורות בהוספה של מחשבים וכיותות מחשוב חדש או צמצום של מספר מחשבים והפיכה של וכיות מחשוב לכיתת לימוד רגילות וכדומה.

המערכת תאפשר לסטודנטים למצוא מחשבים פנויים ברחבי הקמפוס.

המערכת מומשה בטכנולוגיות מתקדמות. הממשק הניהולי וממשק משתמש הקצה הם אינטראקטיביים. ניתן לצפות בתופעות המחשבים בזמן אמת, לעירוך את וכיות המחשב בקלות ולהפיק מגוון דוחות שימושיים המנתחים את פעילויות המחשבים באוניברסיטה ולהציגם בצורה בהירה וגרפית.

במהלך ביצוע הפרויקט השתמשנו בכלים שלמדו במהלך התואר כגון: נושא אפיון, הגדרת דרישות, ניתוח ועיצוב המערכת, פיתוח בסביבה אינטרנטית, תקשורת ואבטחת מידע, עבודה עם בסיסי נתונים, בדיקות תוכנה, ניהול הפרויקט ועוד.

המערכת הוטמעה למעשה בראשת האוניברסיטה בסביבה המדמה את מבנה הרשות ותוצג מול גורמים מנהליים באוניברסיטה לאישור ולפעולת שוטפת באתר האוניברסיטה.

מוגבלות קיימות של המערכת הן :

- מיקום בתחוםי אוניברסיטת בן גוריון ומותקנת רק על שרתיםקיימים באוניברסיטה.
- מספקת שירות בעברית בלבד.
- המערכת לא מותאמת לטלפונים חכמים. עקב אילוצי זמן ותקציב.

בעתיד נרצה להתאים את האתר לטלפונים חכמים, להוסיף תמיכה בשפה האנגלית, להוסיף סוגים נוספים של דוחות, לאפשר גישה מהוуз לראשת האוניברסיטה ולהכניס את המערכת לפעולת שוטפת באתר האוניברסיטה.

23. נספחים

23.1. מערכות תפוצה נוכחות

23.1.1. תצוגות אתר

שיעורים מתוכנים להיוון ומצב התפוצה של כיתות המחשב באוניברסיטה

מספר עדות תפומות בכיתות שעדרם כל 3 דקות
 כיתות ציבוריות (34,35) אוניברסיטה (72) בית האמצעי (15) בית ספר לניהול (15) ציוד רפואי (63) הגדה פינית (59) הנחתה (33) הנחתה לבנות (55) הנחתה מזכירות מדת (90) הנחתה מזכירות מינה (96) הנחתה מזכירות מסחרת (22) הפלטה לדוחות חווית (74) הוירטואלי (72) גזים (29) ללבה (72) גדי חיים (38) גענינג מעוניין (17) גפריה (22) גודה צויאלית (17) גטסה (54) פטילוגיה (98) קאנט שדה בוקר קלפט אלט (16) בפאת תשתיות וזרוי (16)

המחלקה הטכנית	שם עמדת הטכנית	שיעור הכיתה הטכנית לשיעור	שיעורים מtocנים להיוון	מחלקה: כיתה	מחלקות בניין/מקום
המחלקה הטכנית	--	הכיתה הטכנית לשיעור	22:00 - 20:00 16:00 - 15:00 14:00 - 11:00	מערך המחשב כיתות: 310 ציבוריות כיתה	כיתות ציבוריות (35,34)
המחלקה הטכנית	26 (מצוד)	הכיתה הטכנית לשיעור	-- אין --	מערך המחשב כיתות: 35- ציבוריות כיתה- 001	כיתות ציבוריות
המחלקה הטכנית	--	הכיתה הטכנית לשיעור	22:00 - 20:00 16:00 - 14:00	מערך המחשב כיתות: 35- ציבוריות כיתה- 401	כיתות ציבוריות
המחלקה הטכנית	--	הכיתה הטכנית לשיעור	22:00 - 20:00 16:00 - 14:00	מערך המחשב כיתות: 35- ציבוריות כיתה- 402	כיתות ציבוריות
--	--	הכיתה		מערך המחש בזיהוג (35)	

23.1.2. תצוגת מפת כיתה



23.2. מערצת נתוני תפוסת כיתות ציבוריות

23.2.1 מסך כניסה למערכת

נתוני תפוסת כיתות ציבוריות

שם משתמש:	luber
ס.מ.א.:	*****
ת"ז:	
<input type="button" value="המשר"/>	

23.2.2 טבלת ניהול מחשבים

נתוני תפוסת כיתות ציבוריות
מאתאריך: 4 נובמבר עד תאריך: 7 נובמבר שעות: 9:00 - 21:00 ב- 2017

גירסה להדפסה

כיתה	מספר המחשב, בין בית הספרוות כיתה
ט' חנוך[פעלולן]	121
ט' שמש ממען[אוח שימוש ממען]	218
ט' חנוך[פעלולן]	310
ט' שמש ממען[אוח שימוש ממען]	35.001
ט' כהה[ט' כהה]	35.401
ט' כהה[ט' כהה]	35.402
ט' כהה[ט' כהה]	35.403
ט' כהה[ט' כהה]	35.404
ט' כהה[ט' כהה]	555
ט' כהה[ט' כהה]	560
ט' כהה[ט' כהה]	632
ט' כהה[ט' כהה]	75
ט' כהה[ט' כהה]	Aranne ספירה כיתה
ט' כהה[ט' כהה]	Aranne-Class ספירה כיתה
ט' כהה[ט' כהה]	BME-028 סדרה ביזואט, בין כיתה 63
ט' כהה[ט' כהה]	Chem-102 כיתה 29
ט' כהה[ט' כהה]	Ciema, בין כיתה 59 סדרה כימית, בין כיתה 59
ט' כהה[ט' כהה]	ChemEng-237 כיתה 104 הסדרה מושרות הקשורת, בין כיתה 95
ט' כהה[ט' כהה]	CSE-104 הסדרה מושرات קשורת, בין כיתה 95
ט' כהה[ט' כהה]	CSE-105 הסדרה מושرات קשורת, בין כיתה 95
ט' כהה[ט' כהה]	CSE-109 הסדרה מושرات קשורת, בין כיתה 22
ט' כהה[ט' כהה]	Econ-443 מללה, בין כיתה 72

נתוני תפוסת כיתות ציבוריות מטאrik 4 בנובמבר 2017 עד לתאריך 7 בנובמבר 2017 בין השעות

9:00 - 21:00 כל שורה בטבלה מצינית מעבדה.

23.2.3 נתוני תפוסת כיתה

תמונה	ט' שמש ממען[אוח שימוש ממען]	ט' חנוך[פעלולן]	ט' כהה[ט' כהה]	חומר לדף סיכום	גירסה להדפסה
L218W001	5.09	42.92	5.09		[218] 2017-09-00 00:00-21:00 עד תאריך: 7/11/2017 מטאrik: 4
L218W002	5.44	47.88	5.44		
L218W003	6.01	50.17	6.01		
L218W004	5.15	43.82	5.15		
L218W005	5.54	49.20	5.54		
L218W006	6.30	54.24	6.30		
L218W007	6.55	57.71	6.55		
L218W008	5.48	48.33	5.48		
L218W009	6.35	54.93	6.35		
L218W010	5.17	44.03	5.17		
L218W011	4.37	38.51	4.37		
L218W012	3.37	30.14	3.37		
L218W013	2.42	22.60	2.42		
L218W014	5.23	44.97	5.23		
L218W015	3.17	27.47	3.17		
L218W016	3.23	28.26	3.23		

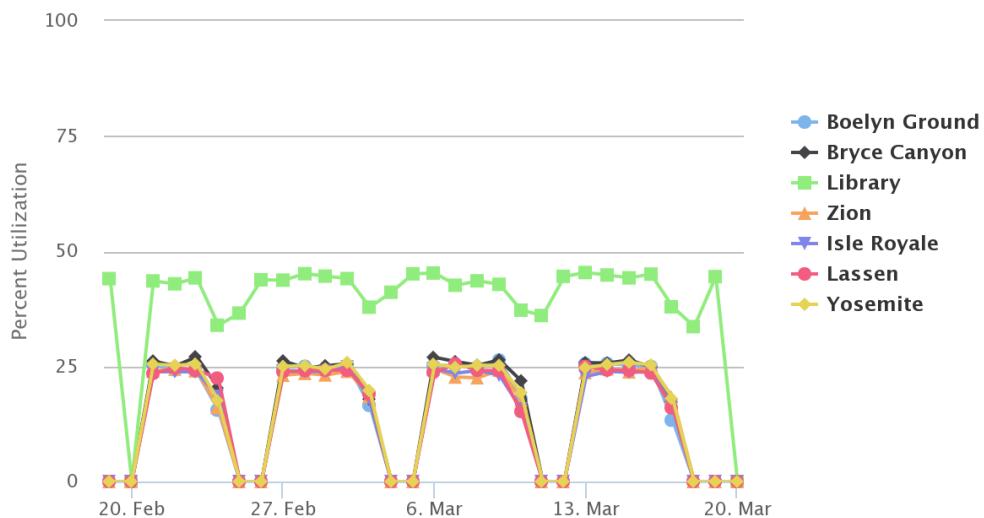
נתוני התחנות של כיתת מחשבים 218. מקשור לטבלה מעלה.

LABSTATS .23.3**.23.3.1 היסטוריה שימוש**

הגרף להלן מתרחץ ציר זמן היסטורי המציג שימוש בתחנות לפי קבוצה,-tag תחנה או קבוצת Tags של תחנות.

Usage History

Show a historical timeline displaying station utilization by group, station tag, or station tag group.

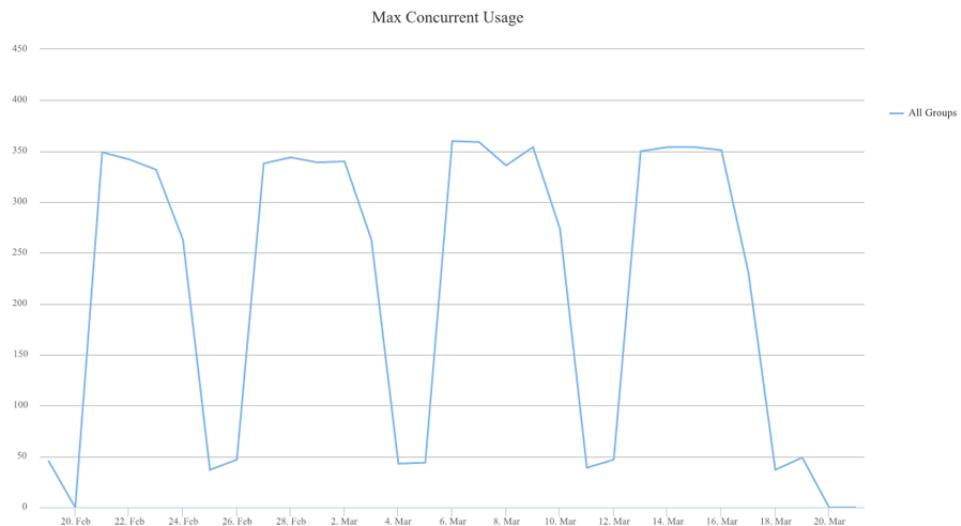
**.23.3.2 שימוש בתוכנות**

הטבלה להלן מציגה סטטיסטיות שימוש מוגברת ביישומים

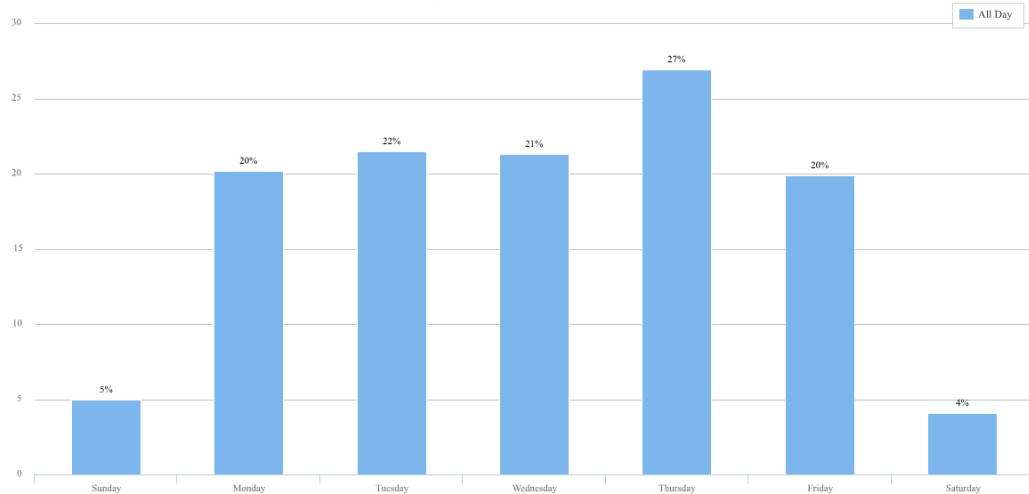
Report Criteria		Results								
Search:										
Application Name	Launch Count	Station Count	Unique User Count	Total Usage (Hours)	Average Usage (Hours)	Focus Time (Hours)	First Launch Time	Last Launch Time		
acrobat	1,629	418	1,358	1,735.0	1.1	867.9	2017-02-19 05:22 AM	2017-03-19 09:51 PM		
Adobe Dreamweaver	1,666	416	1,397	1,692.9	1.0	810.7	2017-02-19 05:35 AM	2017-03-19 01:54 PM		
Adobe Fireworks	1,687	414	1,408	1,759.7	1.0	872.1	2017-02-19 06:13 AM	2017-03-19 08:17 PM		
Adobe Flash	1,613	415	1,360	1,686.8	1.0	851.1	2017-02-19 05:35 AM	2017-03-19 08:08 PM		
Adobe InDesign	1,715	419	1,416	1,749.6	1.0	859.5	2017-02-19 12:01 PM	2017-03-19 06:49 PM		
Adobe Photoshop	1,639	419	1,387	1,643.0	1.0	802.1	2017-02-19 05:19 AM	2017-03-19 07:40 PM		
Amazon	338	229	316	123.8	0.4	127.1	2017-02-21 09:39 AM	2017-03-19 08:20 PM		
Autocad	1,699	413	1,419	1,700.1	1.0	829.9	2017-02-19 05:16 AM	2017-03-19 08:36 PM		
Facebook	370	230	345	132.2	0.4	127.8	2017-02-19 12:04 PM	2017-03-19 06:43 PM		
Final Cut Pro	1,671	413	1,407	1,740.2	1.0	857.8	2017-02-19 05:33 AM	2017-03-19 09:26 PM		
	42,116	n/a	n/a	40,294.6	1.0	21,046.3	2017-02-19 05:05 AM	2017-03-19 10:48 PM		
Showing 1 to 10 of 38 entries		<input type="button" value="First"/> <input type="button" value="<"/> <input type="button" value="1"/> <input type="button" value="2"/> <input type="button" value="3"/> <input type="button" value="4"/> <input type="button" value=">"/> <input type="button" value="Last"/>								

.23.3.3 מקסימום שימוש .23.3.3

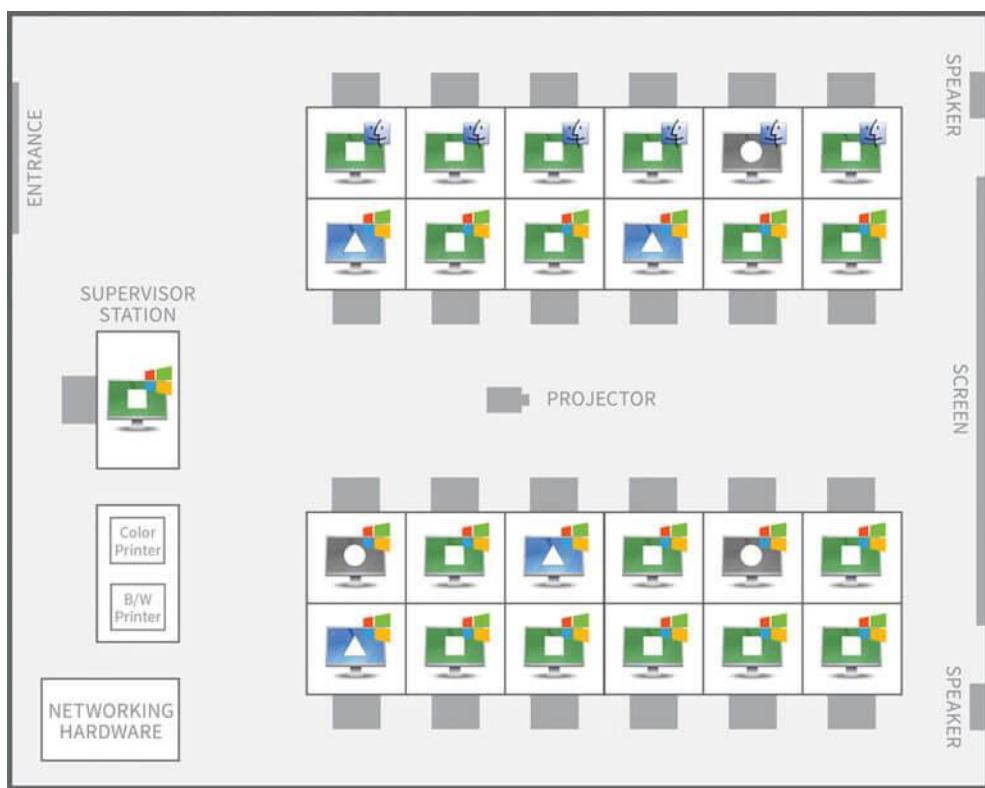
מציג מקסימום שימוש במחשבים מקביל לימי

**.23.3.4 שימוש ממוצע .23.3.4**

Average Weekly Station Utilization



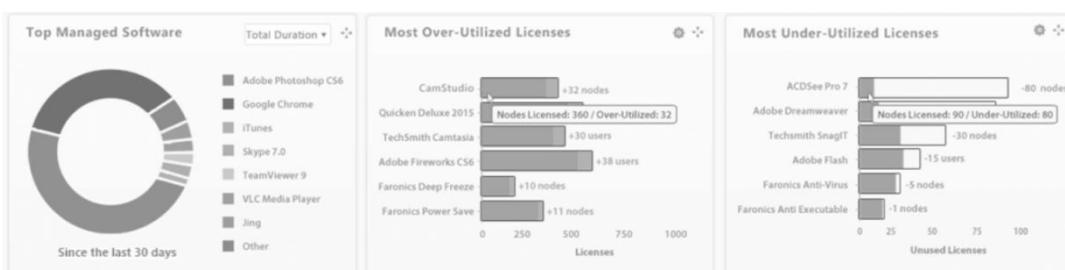
מציג שימוש ממוצע במחשבים לפי יום בשבוע

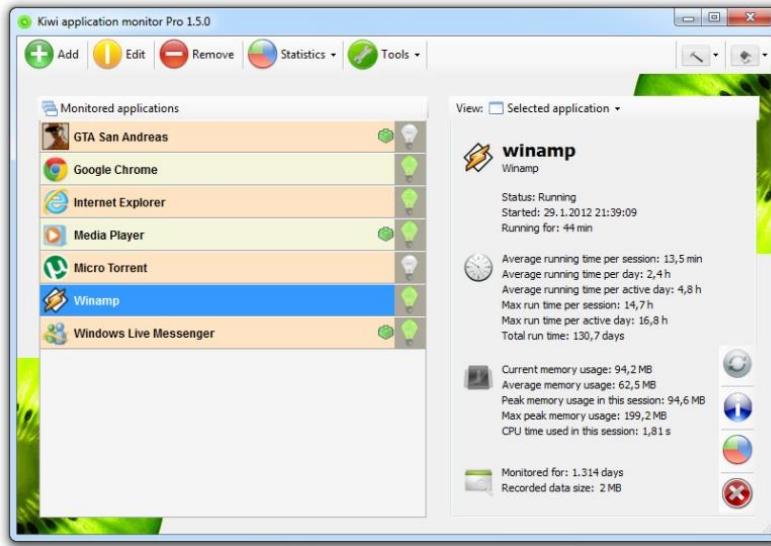
.23.3.5. מפה של מעבדה

ניתן לראות אילו עמדות בשימוש, איך מפוזרים המחשבים בחלל ואיזו מערכת הפעלה יש על כל תחנה.

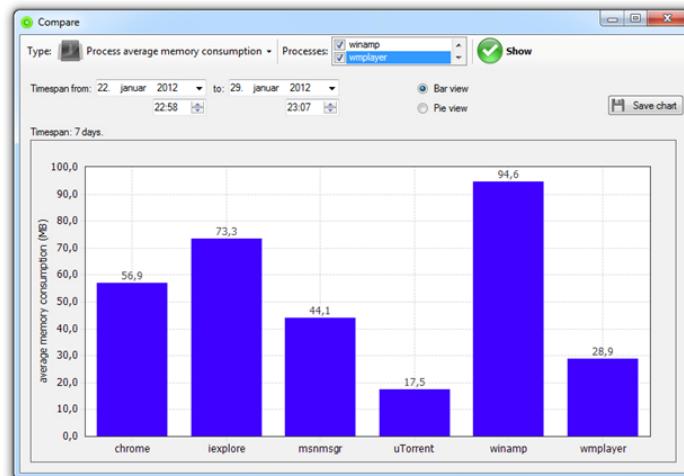
Deep Freeze .23.4**.23.4.1. מספר משתמשים במחשבים לפי תאריך****.23.4.2. שימושיות רישיונות**

התוכנות בעלות מספר הרישיונות הלא מנוצלים ביום, המנוצלים יותר על המידה ביתר, והתוכנות
הכǐ משומשות בחודש האחרון

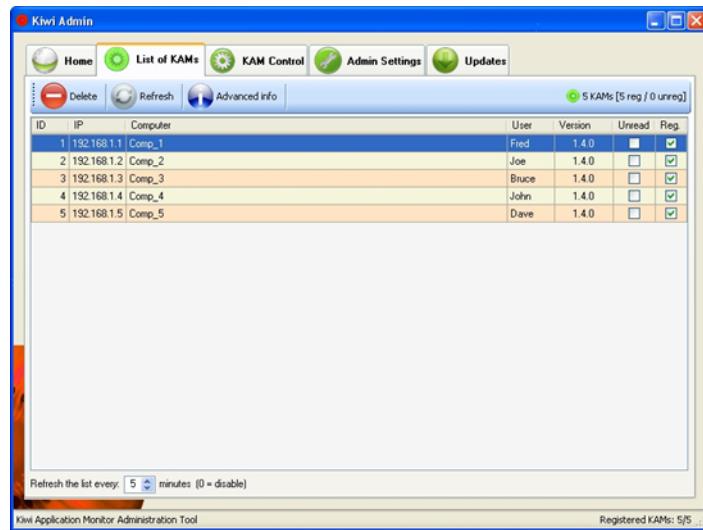


Kiwi Application monitor .23.5**רשימת היישומים הנמצאים במעקב .23.5.1****.23.5.2 שימוש ממוצע בזיכרון לפי יישום**

ניתן לבחור את אופן הציגות ותקופת זמן.

**.23.5.3 רשימת מחשבים תחת פיקוח ה-Kiwi admin**

שולט על מספר מספק (KAMs) Kiwi Application monitor בראשת, גישה להגדרות ופיקוח על המידיע המוצג בכל אחד מהמחשבים ואיסוף המידע למטרות סטטיסטיות.



הודעות שהועברו למשתמש kiwi admin תחת פיקוח.

.23.5.4



24. Summary

We developed a system that displays the current and historical capacity of computers at the public labs at Ben-Gurion University.

The system's requirements were characterized by Mr. Luber, head of Personal Computation Section department. This system provides a solution to all the client's demands and needs. Also, it is easy to use, and well designed. The system collects the university's computer usage data and will be used by the IT administrators, and those who holds management position at the university. Also, the system will use them to attain a deeper understanding about the use of computers and support the decision making, regarding the logistics of computers in the labs (adding or removing computers or classroom, etc.).

The system was implemented by advanced technologies. The software's interfaces are internet based; the user can watch the computer's capacity in real time, so as managing the computer classroom easily, and extract from it useful reports that analyze the activity of university computers.

During the work on this project, we used tools that we acquired during our studies such as: System characterization, requirements settings, system analyzing and designing, internet environment developing, data security and communication, databases, project management and so on.

The system was assimilated at the university network, in a setting that simulate the network structure, while waiting for approval to be implanted in the production environment.

The system has several limitations: a. It can be access only in the university domain. B. It provides service only in the Hebrew language. C. It is not adapted to smartphones.

During the work on this project, there were several challenges that arose. Examples for some of them are concerning with permissions, foreseeing all possible scenarios and outcomes, teamwork, time management and designing the system.

Surely, the execution of this project was very beneficial to us, in the sense that it provided us with integration of all the knowledge and skills we acquired during our studies, and the opportunity to implement it in real life.

25. Executive Review

The public computers at the university are expensive resource, manpower wise and equipment wise. The decision making regarding these resources is being made while taking into consideration the number of students at every faculty, the number of courses and classes that includes the use of the computers and more.

Due to technological changes (mostly cloud services that enables users to use "heavy" software on their private computers), less and less students are using the public computers at the university. Nevertheless, there is still a big need for these computers- for students' exercises, and for frontal lectures. As of today, there are 1200 public computers at the university. For a useful exploitation of these resources, a tracking of their usage is much needed.

The responsible body of the public computers is the Computation Center Department of the university. As of today, there are two different software that concern these computers; "public classroom capacity" system, a system that displays the scheduled classes and the classroom capacity of everyday, and "public classroom data" system, that displays the average time of use in the classroom's computers in a defined time range. These two systems are old and have many limitations such as lack of adaptation to newer systems, slow reaction time, many bugs and more. Over the years, several attempts were made to improve and fix the existing software, but current technologies could not support it, so as the lack of cohesiveness that the code was written in. these systems fail to provide reliable information, and they are almost unusable.

Our suggested solution is a computer's activity monitoring system. This system uses new technologies, and provides the display of computer capacity over the university campus, so as a management system that enables the extraction of reports that provides information of the public computer's activity.

Started working on this project, while meeting with head of Personal Computation Section, Mr. Michael Luber, we assessed the current situation, so as the expectation of the system, its limitation and constraints. Also, we define the system needs and examined possible alternatives. From reviewing the existing systems, we concluded that they do not provide with the fundamental needs of the required system, and that the right course of action would be developing a computer's activity monitoring system.

After that we examined different technologies and developing methodologies to develop the system. We chose to use ASP.NET MVC working environment because it met with most of the defined criteria, and we have the most working experience with it. In addition, we chose to work with Microsoft SQL data base engine because of our extensive experience with it, but also because it is ideal for the current working environment. As to our developing methodology we chose Unified Process. Although we don't have working experience with this methodology, it was the most suitable out of all the ones we examined. This methodology emphasizes on understanding the requirements and needs early in the process, so the development process start when its goals are well clear and defined. Additionally, the

development starts from the most critical and main parts of the system and then expands through usage scenarios, a feature that is well suited to our project.

At the next stage we analyzed and designed the upgraded system, defined the function and requirements that the system will include, while emphasizing on its ease to use and usefulness. To understand the structure of the future system, we created usage scenarios, data flow diagram and division charts. We put a lot of effort into this part, as it is very meaningful part of the system development lifecycle. Also, there have been a meticulous analyze to reduce the number of faults and obstacles in the future.

After this phase, we approached the software implementation. We divided the development to iterations, so that in every iteration we focused on a different usage scenario, inspired by the development methodology that we chose. The development in the MVC architecture enables us with a clear, comfortable division of the work, so that one of us focused on developing the display layer, and the other one focused on developing the model layer.

After most of the development was completed, we asked Mr. Luber to examine the system, to see if it meets his expectations. Mr. Luber showed his approval of the system, and gave us more emphasis and requirements. Additionally, to promote the testing and assimilation phase, he allocated resources to build a testing environment that will be as similar as possible to the production environment in the university network.

After installing the required systems, defined and adjusted the allocated resources, we started the process of extensive testing. We checked that all the system parts are working well together and examined the reliability of the data collection as well as the system's statistical analyses.

As for today, the computer's activity monitoring system is a whole system that supply all of client's demands and needs as characterized. The system collects the usage data of the university's computers. Both interfaces are used in the internet. In its configuration, the system is easy to use, and well designed. The system enables the user to observe the computer's capacity in real time, so as managing the computer classroom easily, and extract from it useful and graphical reports that analyze the activity of university computers.

In the future, this system will be introduced to administrative factors in order to get approved and to be implanted into university site.