

מחלקת הנדסת תעשייה וניהול

**פרויקט בקורס " חקר ביצועים 2 "**

בהנחיית ד"ר גבי פינטו

מתרגל מר אמיר ברודנר

**מגישים:**

חן שליו 203205984

יונתן עטיה 311624381

נועם אמרון 305568818

מיכאל יירמולייב 311461263

תאריך הגשה: 12.06.2018

# **תקציר פרויקט**

במסגרת הקורס "חקר ביצועים 2" בחרנו בפרויקט אשר מתבסס על בעיית "תרמיל הגב" בשילוב של תכנון לינארי, שם הפרויקט- "Optimize-IT".

במסגרת התואר "הנדסת תעשייה וניהול", נתבקשנו לפתח/ לשדרג/ להריץ מערכת מידע שפותרת בעיה ע"י שימוש באלגוריתם. לכן החלטנו לפתח אפליקציה לחישוב זמני למידה לכל מבחן בתקופת המבחנים של הסטודנטים. מטרתו לתכנן את זמני הלימוד לכל מבחן בהתאם לפרמטרים שהוכנסו על ידי המשתמש בכדי לעזור לסטודנטים להיערך בצורה יעילה ומועילה יותר לבחינות.

כתיבת האלגוריתם נעשה לפי בעיית תרמיל הגב בשילוב תכנון לינארי אשר נלמדו במהלך לימודי התואר.

"בעיית תרמיל הגב"- בבעיה זו נתונה קבוצת עצמים שלכל אחד מהם משקל וערך, ובנוסף נתון חסם על הערך. המטרה היא למצוא אוסף של העצמים הנתונים שסכום משקליהם אינו עולה על החסם הנתון, ומחירו מרבי.

אצלנו בפרויקט, בנינו את פונקציית המטרה כך שננצל ניצול מרבי של הזמן למטרת למידה למבחן בעזרת פרמטרים אשר מקבלים מהמשתמש: מספר מבחנים, מספר נקודות זכות לכל מבחן ועוד. ישנם גם מספר אילוצים כמו: מספר שעות ביום אשר ניתנות ללמידה (יום העבודה שהוגדר 8 שעות), האם הקורס חוסם אופציה לקורסים נוספים ועוד.

האפליקציה פותחה בעזרת Android studio וכרגע זמינה למשתמשי אנדרואיד בלבד (בהמשך נוכל לפתוח את האפשרות למשתמשי IOS). האפליקציה משתמשת בחישובי רקע אשר המשתמש לא נגיש אליהם ורק יכול להזין נתונים בכדי לאפשר לאפליקציה לבצע את החישובים ולפלוט את הפלט הסופי למשתמש. לצורך כך אנו נעזרים במדריכים מקוונים, בעצות ממתכנתים ובעיקר מניסוי ותהייה. אנו מרצים מודלים ניסיוניים בטאבלט אשר משמש כסימולטור לתקינות האפליקציה ולאיתור וניתור הבאגים.

לצורך כך נערוך שאלון קצר ומידי בין משתמשים שונים העוזרים לנו לאמוד את חוויית המשתמש ולקבל הערות בכדי לאפשר לנו לדעת היכן אנו צריכים לשפר/לתקן חלקים שונים באפליקציה ועל ידי כך לקבל מידע אמין על פתרון האלגוריתם.

**תוכן עניינים**

[**תקציר פרויקט** 1](#_Toc516420873)

[**מבוא** 3](#_Toc516420874)

[**נושא הפרויקט** 4](#_Toc516420875)

[**הבעיה והאלגוריתם הנבחרים** 4](#_Toc516420876)

[**על המערכת** 5](#_Toc516420877)

[**תהליך בניית המערכת** 5](#_Toc516420878)

[**מדדי ביצוע** 6](#_Toc516420879)

[**סיבוכיות הקוד** 6](#_Toc516420880)

[**בחינת המערכת** 6](#_Toc516420881)

[**איכות הפתרון** 6](#_Toc516420882)

[**תכנות-בדיקה-הטמעה** 7](#_Toc516420883)

[**ניהול הפרויקט** 7](#_Toc516420884)

[**חלק התכנות** 7](#_Toc516420885)

[**בדיקות תוכנה** 8](#_Toc516420886)

[**הטמעה** 8](#_Toc516420887)

[**סיכום מסקנות והמלצות להמשך** 9](#_Toc516420888)

[**סיכום ומסקנות** 9](#_Toc516420889)

[**המלצות להמשך** 9](#_Toc516420890)

[**ביבליוגרפיה** 10](#_Toc516420891)

[**נספחים:** 11](#_Toc516420892)

[**נספח 1: פתירת הבעיה בעזרת תכנון דינאמי דטרמיניסטי באופן ידני (הדגמה)** 11](#_Toc516420893)

[**נספח 2: תמונות מקוד התכנות** 11](#_Toc516420894)

[**נספח 2.1: כניסה למערכת** 11](#_Toc516420895)

[**נספח 3: צילומי מסך – דרכי תקשורת בין חברי הצוות** 11](#_Toc516420896)

[**נספח 3.1: WhatsApp** 11](#_Toc516420897)

# **מבוא**

הפרויקט שלנו במסגרת קורס חקר ביצועים 2, עוסק בפיתוח מערכת מידע שתהווה מערכת תומכת החלטה עבור המשתמש. המערכת עושה שימוש באלמנטים מאלגוריתמים ידועים שלמדנו בקורס לרבות תכנון ליניארי (במסגרת קורס חקר ביצועים 1 וכן חקר ביצועים 2) וכן תכנון דינאמי, כאשר השתמשנו במסגרת תכנון דינאמי ברעיונות ואלמנטים מ"תרמיל הגב" כמו גם מתכנון דינאמי קלאסי.

כסטודנטים אנו מודעים לקשיים הרבים שבניהול זמנים, ובתעדוף נכון של המשימות האקדמאיות, הן בתקופת הסמסטר הפעיל והן בתקופת המבחנים. אנחנו בחרנו להתמקד בבעיה ידועה (בעינינו) המתייחסת לתכנון נכון של זמני הלמידה בתקופת המבחנים.

הבעיה העיקרית היא שכאשר כבר מגיעה תקופת המבחנים אין לסטודנט פנאי להשקיע זמן בתכנון ניהול הזמן שלו בצורה מועילה ויעילה[[1]](#footnote-1). מעבר לכך, הוא צריך להחליט כיצד לתעדף נכון את זמן הלימוד בכל קורס. אנחנו מתמקדים בסביבת לימודים עבור סטודנטים להנדסה, כאשר התבססנו על העבודה שיש מרווחים צפופים יחסית בין מבחנים, לפעמים אפילו עד יום הפרש בין מבחנים. עובדה זו מקשה על הסטודנט ללמוד בצורה יעילה לכל מבחן.

מערכת המידע שלנו, נועדה לתמוך בהחלטתו של הסטודנט בחלוקת הזמנים של הלימוד בתקופת המבחנים. מערכת המידע שלנו, מתבססת על מספר פרמטרים שיסופקו על-ידי המשתמש, במקרה שלנו סטודנט. לפי פרמטרים אלו, נחשב כמות שעות סה"כ נדרשת ללימוד למקצוע מסוים, נתן למשתמש פיצול ברמה היומית – כלומר הוא יהיה מסוגל לדעת ביום מסוים לכמה קורסים שונים הוא לומד, וברמה התת-יומית הוא יקבל פלט האומר לו כמה מתוך 8 השעות האפשריות ללימוד יושקעו בלימוד לכל אחד מהקורסים אליהם ילמד באותו היום.

בנוסף לאפליקציה פיתחנו מערכת כניסה ורישום של משתמשים (באמצעות שרת MySQL, מערכת שמירת מידע בשרת חיצוני ובמעטפת על האפליקציה בשם SharedPreferences, וזאת על מנת שתהיה שמירת היסטוריה והמשתמש יוכל לדעת בכל זמן נתון מהי התכנית הלימודית שלו.

בפרקים הבאים נציג את האלגוריתם ואת מאפייני התוכנה, נסביר על האילוצים הקיימים וכיצד בנינו את החישובים הנדרשים ונסביר את השימוש בתכנה. נפרט באיזו שיטה בחרנו לניהול הפרויקט וביצועו, נציג את חלוקת העבודה בין כל חברי צוות הפרויקט. כמו כן, נציג את האלגוריתם שנכתב בשפות התכנות השונות (Java, XML, JSON, PHP, Android), נציג גם את התוצר הסופי, נסביר על הקשיים, השגיאות והבעיות שנתקלנו במהלך העבודה, כיצד התמודדנו וכיצד הם נפתרו ולבסוף נציג את מסקנותינו והמלצותינו להמשך.

# **נושא הפרויקט**

## **הבעיה והאלגוריתם הנבחרים**

הבעיה בה בחרנו להתמקד היא מקרה פרטי של בעיית תכנון זמנים. הבעיה שלנו נמצא בסביבת תכנון זמנים, הרגיש לדד-ליין, משקל הפרויקט עצמו, והן רגיש לעובדה כי מדובר בריבוי פרויקטים.

כאשר אתה סטודנט, ועליך לבצע ריבוי "פרויקטים" (במקרה שלנו מבחנים) במרווחי זמן קצרים, הפרשי הזמן בין פרויקט לפרויקט לא מאפשרים לך להתכונן לפרויקט הבא בצורה אופטימלית. לכן הגענו להבנה כי על מבצע הפרויקטים (במקרה שלנו הסטודנט) לבצע מספר פרויקטים במקביל.

משהבנו כי על הסטודנט ללמוד למספר מבחנים במקביל עלו עוד כל-מיני שאלות בדרך. לכמה מבחנים יוכל סטודנט ללמוד ביום ? ואיך נקבע עדיפות למבחן אחד על פני האחר באותו היום, במסגרת השעות האפשריות (8 שעות שנקבעו כפי שיוסבר בהמשך).

מורכבות הבעיה, אינה מאפשרת פתרון פשוט. ניחנו בכך כאשר אחד מחברי הצוות עבר סמסטר ובו היה עליו לגשת לכל מועדי ב', זאת לאחר שניגש למועדי א', עקב תכנון לקוי של תכנית הלימוד שלו למבחנים השונים. דבר זה עלול לגרום למצב שבו הסטודנט "מותש" וזהו לא אפקט רצוי של תקופת המבחנים. לכן, בנוסף לאפשרות לחשב ולתכנן נכונה את הזמנים שלו, המערכת תאפשר לסטודנט להגיע למצב הרצוי של מנוחה בין הסמסטרים.

בחרנו לבנות מודל משל עצמנו, השואב אלמנטים מתוך תכנון ליניארי, תכנון דינמי וכן מקרה פרטי של תכנון דינמי והוא "תרמיל הגב". מהתכנון הליניארי שאבנו את האילוצים שהם במקרה שלנו אילוצי עשות אפשריות ביום, וכן האם מדובר בקורס קדם ואז הוא חוסם את הקורסים מהסמסטרים הבאים. מתכנון דינאמי שאבנו את התלות של החלטה אחת בהחלטה שניה, ולכן כל קביעה של האפליקציה לגבי שעות לימוד בנושא מסוים משפיעה על הקביעות הבאות בנושאים הבאים. מתרמיל הגב שאבנו את עניין המשקל של כל קורס, לצורך קביעת תעדוף ברמה היומית. בנוסף קבענו כי המערכת תתחשב בתאריכים של הקורסים השונים כדי לתת קביעות לקורס אחד על פני האחרים, לאחר שהוא נקבע כקורס החשוב, וניתן לו משקל, אותו הדבר קורה לגבי יתר הקורסים שיש ללמוד אליהם באותו היום.

על מנת להשתמש בכל האלגוריתמים הללו, פיתחנו מודל תיאורטי . המודל מתייחס לעובדה שיש מרווח זמן קצר בין פרויקט לפרויקט[[2]](#footnote-2), ולכן כמות השעות האפשרית להשקיע בפרויקט הבא איננו מספיק. במקום זאת, בהתבסס על משקל הפרויקט ופרמטרים נוספים, מחשבים שעות נדרשות להשקעה באותו פרויקט. מאותו רגע, מסמנים ביומן התאריכים את יום הסיום. ומושכים ממנו אחורה. בכל יום שקודם ליום הסיום, נסמן כמות שעות מסוימת מאותו פרויקט, ונחסיר מסך השעות הנדרשות להשקעה בפרויקט, כאשר תנאי העצירה הוא שאין יותר שעות נדרשות. כמות השעות המסומנת בכל יום, מחושבת על פי משקל הפרויקט. כאשר אותו תהליך קורה עם הפרויקט הבא, ייתכן וייווצר מצב של חפיפה בין פרויקטים בהם יש להשקיע באותו יום, לכן במקרה וכבר סומן באותו היום פרויקט מסוים, נערכת השוואת תאריכים ומשקלים כדי לראות מי קודם למי, והפרויקט השני יקבל שעות מתוך השעות האפשריות באותו היום הנותרות. וכך הלאה.

## **על המערכת**

המערכת שלנו פועלת בפלטפורמת Android שמורכבת משפות שונות: Java, PHP, JSON, Android, XML. השתמשנו במסד נתונים MySQL לשמירת נתוני משתמשים.

**נתוני קלט** – הסטודנט מזין במערכת עבור כל קורס: את תאריך הבחינה, מסמן האם מדובר בקורס קדם, ואת מספר נקודות הזכות. הוא גם מכניס את שם הקורס, אך לפרמטר זה אין חשיבות רבה עבור המערכת.

**עיבוד** – המערכת שומרת את כל הנתונים של המשתמש, על סמך הנתונים מחשבת עבור כל קורס משקל הקורס, כמו גם קדימות הקורס, וגם את שעות הלימוד הנדרשות (המלצה שלנו על בסיס חישוב רקע). לאחר מכן, המערכת מחשבת בכל יום נתון כמה שעות על הסטודנט להשקיע בלימוד לכל קורס רלבנטי. המערכת עושה זאת בצורה איטרטיבית על כל אחד מהקורסים, עד אשר כל התנאים מוצו ומולאו.

**פלט** – המערכת מציגה למשתמש תכנית עבודה[[3]](#footnote-3), שהביטוי שלה כרגע הוא רשימה של פרויקטים עם המאפיינים של כל מבחן, כמות שעות נדרשת ללימוד, תאריך התחלת הלימוד ותאריך הסיום, וכן האם מדובר בקורס קדם או לא.

### **תהליך בניית המערכת**

השלב ההתחלתי היה לנתח כיצד ניתן לחשב את כמות השעות אשר על המשתמש להשקיע על מנת שלכל הפחות יעבור את המבחן. מניסיון העבר של חברי הצוות, ראינו ככל שמספר נקודות הזכות גדולות יותר כך היקף החומר הינו גדול יותר ו/או קשה יותר, כך גם לגבי קורסים שהינם חוסמים. בנוסף נלקח בחשבון ממוצע זמן עבודה אפשרי של הסטודנט שעומד על 8 שעות ביום. זמן ממוצע זה מתבסס על מספר חוקים אמפיריים, חוק ירקס – דודסון [[4]](#footnote-4) וחוק הופשטטר[[5]](#footnote-5), בנוסף בחישוב זמן זה התחשבנו במציאות של חיי סטודנט ממוצע, כלומר, זמן עבודה, נסיעות, שינה וכדומה.

פונקציית חישוב זמן למידה / המטרה :

*האם קורס n חוסם או לא- אם כן = 1, אם לא = 0*

*מספר נקודות הזכות של קורס n*

*מספר הקורס.*

## **מדדי ביצוע**

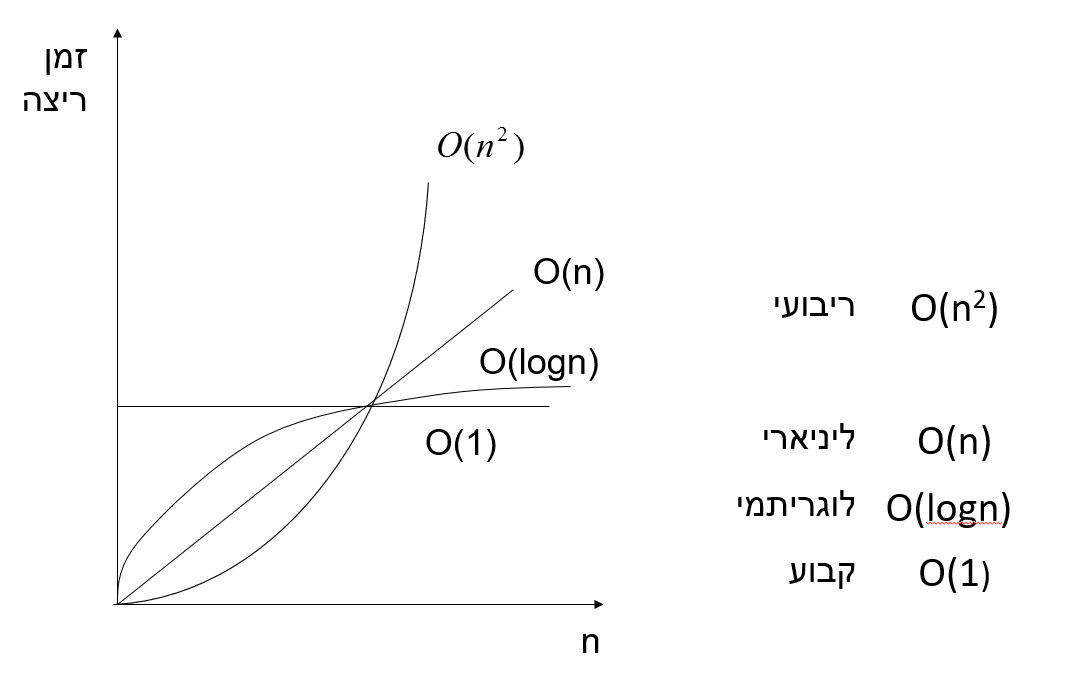
### **סיבוכיות הקוד**

הסיבוכיות באלגוריתם שלנו מורכבת ממספר גורמים כמו: הוספת משתמש, חיפוש משתמש קיים, חישוב זמן למידה, הוספת מבחן ועוד.

בכלליות סיבוכיות זמן ריצת האפליקציה הינו O(n) כאשר נפרט על סיבוכיות ומרכיבי האלגוריתם:

* חיפוש משתמש קיים /הוספת משתמש חדש - סיבוכיות של O(n) .
* הוספת מבחן חדש – סיבוכיות של O(1) .
* הצגת כל המבחנים- סיבוכיות של O(1).
* חישוב זמן הלמידה (חישוב האלגוריתם) - סיבוכיות של O(n).
* חישוב עדכון פרטי מבחן – O(1).

בחישוב סיבוכיות כללית מתעלמים מקבועים וזמני ריצה קביעה } O(1) } ולכן, הסכום של זמן הריצה של כלל הפונקציות באפליקציה הוא: 2O(n) + 3O(1) = 2O(n) = O(n)



### **בחינת המערכת**

בחנו את המערכת גם בעזרת תוסף לדפדפן בשם Postman אשר בודק כשירות קידוד PHP יחד עם השרת localhostכדי לוודא קישור פועל בין מסד הנתונים לאפליקציה, בנוסף בחנו את המערכת בעזרת מכשיר ייעודי (LG g3 beat) אשר הריץ את קוד התכנות לוודא תקינות של האפליקציה ללא קריסות או נתונים שגויים ובנוסף בדקנו גם באמצעות אמולטור ייעודי המותקן בסביבת העבודה Android Studio לוודא ניקיון מבאגים שונים. יתר על כן שלחנו את קובץ ההתקנה בין מכרים ומשפחה כדי לקבל פידבק ולבחון את נוחיות השימוש.

### **איכות הפתרון**

גרסת המודל הנוכחית שפיתחנו פותר את כלל הבעיות בצורה נכונה ומדויקת בקרוב של 97% מהמקרים שדגמנו, כאמור המערכת בגרסתה הנוכחית מבוססת על אלגוריתם לינארי.

# **תכנות-בדיקה-הטמעה**

## **ניהול הפרויקט**

ניהול הפרויקט התבצע בעזרת שילוב ידע וכלים של כל חברי הקבוצה אשר שותפו באמצעות אמצעים טכנולוגים[[6]](#footnote-6) ובמהלך פגישות שנקבעו בין חברי הצוות במהלך העבודה על הפרויקט, כמו כן נעזרנו ביעוץ והכוונה של דוד סלמון עתודאי להנדסת תוכנה שנה ג' .

כל אחד מחברי הצוות תרם את חלקו בחישוב הידני וגם בחישוב הטכני ובבניית האפליקציה בתוכנת Android studio. כחברי הקבוצה היה חשוב לנו שלמשתמש תהיה חוויה קלילה ופשוטה לשימוש באפליקציה ובעיקר לא לנסות "להיות מקוריים מידי" מבחינה עיצובית, לכן החלטנו על עיצוב פשוט להבנה וקל לתפעול.

**חן שליו-** תכנות האפליקציה והכנת דו"ח

**יונתן עטיה-**תכנות האפליקציה

**נועם אמרון-**תכנות האפליקציה והכנת מצגת

**מיכאל ירמולייב-**הכנת מצגת

## **חלק התכנות**

לאחר גיבוש הדעה אודות נושא הפרויקט, קיבלנו החלטה לכתוב קוד באופן עצמאי, על סמך ידע קודם אשר רכשנו במהלך קורסי התואר בשפתJava, PHP וגם התנסות בסביבת עבודה (פלטפורמה) ללא ידע קודם Android Studio וכן קבלת יעוץ ממתכנת (הוזכר בסעיף ניהול הפרויקט).

תחילה, רכשנו ידע בפתרון בעיות בתכנון דינאמי ובבעיית תרמיל הגב בקורס "חקר ביצועים 2" וגם ידע בפתרון בעיות בתכנון לינארי בקורס "חקר ביצועים 1" על מנת שנוכל לפתור את הבעיה בצורה האופטימלית. ערכנו סימולציה ידנית לבעיה, בעזרת ניסוח משתנים, אילוצים והגדרת "פונקציית המטרה". לקחנו בחשבון שלסטודנט הממוצע לאחר קיזוז שעות שינה, אוכל והפסקות (ללא זמן עבודה) יש בממוצע 8 שעות אשר בהן הוא יכול ללמוד, לפי כך תכננו את החישוב באפליקציה.

לאחר שהתגבשה דעתנו על אלגוריתם הפותר את החישובים הרצויים, רשמנו את שלבי הפעולות הנעשות בהרצה ידנית ותרגמנו אותן לקוד תכנות[[7]](#footnote-7) , שילבנו זאת יחד עם גוף הקוד אשר כולל את המשתנים, האילוצים ו"פונקציית המטרה". בנוסף, פיתחנו מערכת רישום וכניסה של משתמשים באמצעות השפות PHP, Json , MySQL בשילוב Android Studio.

יתר על כן, ביצענו חיבור מהאפליקציה למסד הנתונים על מנת לשמור נתוני משתמש (שם משתמש וסיסמה וכדומה) וגם שלבנו מעטפת של שמירת נתונים של האפליקציה בשם SharedPreferences.

נערכו בדיקות הרצה לוודא ניקיון מבאגים (תקלות) בתכנית. לאחר שהצלחנו להריץ את האפליקציה באופן סופי, החלטנו להעלות את קובץ ההתקנה (apk) לאתר אינטרנט.

בתהליך התכנות נתקלנו בקשיים שונים. הקושי הראשוני היה מציאת הדרך להעביר לשפת תכנות את האלגוריתם שחושב ופותח ידנית לשפת תוכנה, את המשתנים, את האילוצים והגדרת "פונקציית המטרה". קושי נוסף היה למידת והכרה של פלטפורמת Android והפעלה ושימוש בסביבת העבודה שנקראת Android Studio.

הקושי השלישי איתור באגים, שגיאות ריצה, שגיאות פלט. לכן השתמשנו במכשיר ייעודי ובאמולטור המובנה בסביבת העבודה אשר שימשו לשם הרצה, פתרון השגיאות השונות ועוד.

## **בדיקות תוכנה**

בדיקות התוכנה (Software Testing) מהוות חלק חשוב מאוד בפיתוח מערכת. מטרות הבדיקה היא לוודא שהמערכת עונה על הדרישות אשר הצבנו בתחילת הפיתוח ואינה קורסת בהפעלת חלק מסוים או מכל סיבה. כאשר החלטנו לתכנת את האלגוריתם הצבנו מספר מטרות, בראש ובראשונה החלטנו כי המטרה החשובה ביותר היא לבנות מערכת שתעביר למשתמש מידע אמין ומדויק, שתהיה נגישה למשתמשים ותעניק חווית שימוש נוחה.

על מנת לבדוק שעמדנו ביעדים אלו שהצבנו ערכנו מספר בדיקות ע"י הרצת המערכת לאחר סיומה:

בדיקות פונקציונליות - (Functional) לאמת שמערכת אכן מבצעת פעולות לוגיות ואריתמטיות כפי שציפינו. הבדיקה זו נעשתה בעזרת Logcat בו ניתן לראות את כל פעולות הג'אווה באופן ישיר.

בדיקות שימושיות (Usability) - בדיקות נוחות השימוש, יעילות ונגישות העיצוב של האפליקציה. באפליקציה שבנינו השתדלנו שהעיצוב יהיה מנימליסטי אך נעים ויזואלית. בנוסף, רוב המידע מרוכז בעמוד אחד אשר מרכז את רוב המידע ואפשרויות הפעולה של המשתמש. יש חלון בודד להכנסת המבחנים לשימוש המשתמש.

## **הטמעה**

לאחר פיתוח האפליקציה הפצנו בקרב מכרים ומשפחה את קובץ ההתקנה (apk) על מנת שיוכלו לתת את חוות דעתם על האפליקציה מכל היבטיה: נוחות, נראות ואיכות.לאחר בדיקה בקרב המשתמשים שהופצה אליהם ההתקנה, עשינו הערכה כללית לגבי כל הפרמטרים הנמדדים והפקנו לקחים ושדרגנו את האפליקציה מפעם לפעם לפיהם.

# **סיכום מסקנות והמלצות להמשך**

## **סיכום ומסקנות**

ראינו את מאפייניה של בעיית תכנון הזמנים, חקרנו אותה. על סמך בעיה זו, קבענו באילו אלמנטים אלגוריתמים מתוך אילו מודלים להשתמש (סטוכסטי/דטרמיניסטי).

לאחר סיעור מוחות וניסיון לנסח את הבעיה, להתמקד על מה אנחנו רוצים לפתור ואיך, מידלנו רעיון תיאורטי לגבי אפשרויות לחלוקת זמן, בתוך הזמן האפשרי ולא ברמה היומית אלא הרחבנו את ההסתכלות על זמן אפשרי לרמה השבועית ולפעמים דו-שבועית.

למדנו להשתמש בפלטפורמת אנדרואיד סטודיו, למדנו כיצד לקשר בין מערכת מבוסס שפת JAVA ו-XML באמצעות שימוש ב-JSON parsing על מנת לאפשר תקשורת בין הקוד שלנו בJAVA לדפי PHP המשמשים בתורם להעברת נתונים אל ומבסיס נתונים רלציוני (שסופק לנו על-ידי המכללה בפלטפורמת cpanel, והוא mySql).

לסיכום, ראינו כיצד ניתן להגיע ממצב כאוטי בו אנחנו מבזבזים זמן יקר ולא מבוטל על בניית תכנית לימודים, לכך שבאמצעות מערכת פשוטה אנחנו יכולים לפתור לא רק את הבעיה של עצמנו אלא להציע פתרון טוב יותר עבור כלל הסטודנטים המתחבטים ומתקשים באותם מצבים. בסופו של דבר זה יוביל לאופטימיזציה לא רק של יכולת התכנון, אלא גם של הלמידה של הסטודנט, דבר הממקסם את הפוטנציאל של הסטודנט.

## **המלצות להמשך**

אנו בעתיד מעוניינים לשפר את המערכת על מנת שתוכל לתת פתרונות יעילים וממוקדים יותר, כגון: חלוקת זמנים לכל יום באופן פרטני, התחשבות בהפסקות וכו'. כמו כן, לאפשר למשתמשי IOS (iPhone users) להשתמש באפליקציה בפלטפורמה מיועדת ומותאמת. בנוסף, בעתיד אנחנו מתכוונים לאפשר גנריות מסוימת עבור מגוון רחב יותר של משתמשים. המטרה הבאה שלנו היא הוספת יומן שיראה את תכנית העבודה בצורה ויזואלית (נוח יותר למשתמש). כחלק מתכנית האב אנחנו שואפים להביא אפליקציה זו למצב גנרי (בעתיד הרחוק) לאחר שנרכוש את הידע הרצוי, כך שמשתמש הקצה לא יהיה סטודנט אלא כל מהנדס או איש עסקים או מנהל שלה/לו פרויקטים מרובים בסביבה רגישה לזמני סיום.

# **ביבליוגרפיה**

1. <https://www.w3schools.com/> - מאגר מידע לשות PHP, SQL, JSON
2. [www.getpostman.com‏](http://www.getpostman.com) - הורדת תוסף לדפדפן Chrome, לבדיקת קידוד בשפת PHP
3. <https://developer.android.com/> - מאגר מידע למפתחים בפלטפורמת אנדרואיד וXML
4. <https://docs.oracle.com/javase/7/docs/api/> - מאגר מידע למפתחים בשפת JAVA
5. The Relation of Strength of Stimulus to Rapidity of Habit-Formation, Robert Yerkes & John Dodson , *Journal of Comparative Neurology and Psychology*, *18*, 459-482.
6. Gödel, Escher, Bach: An Eternal Golden Braid, Douglas Hofstadter, 1979
7. The Effects of Student Syndrome, Stress, and Slack on Information Systems Development Projects, D. C. Smith University of Cape Town , South Africa

# **נספחים:**

## **נספח 1: פתירת הבעיה בעזרת תכנון דינאמי דטרמיניסטי באופן ידני (הדגמה)**

**ניסוח הבעיה:**

מספר הקורסים n=2 לכן עלינו לרוץ פעמיים (לכל מבחן ) בעזרת הפונקציה.

(פעם ראשונה) מבחן "חדוא 1" תאריך בחינה: 30.06.2018 , נ"ז: 5 , קורס חוסם = 1.

(פעם שניה) מבחן "הנדסת אנוש" תאריך בחינה: 21.7.2018 נ"ז: 4 , קורס חוסם = 0.

**חישוב הידני**

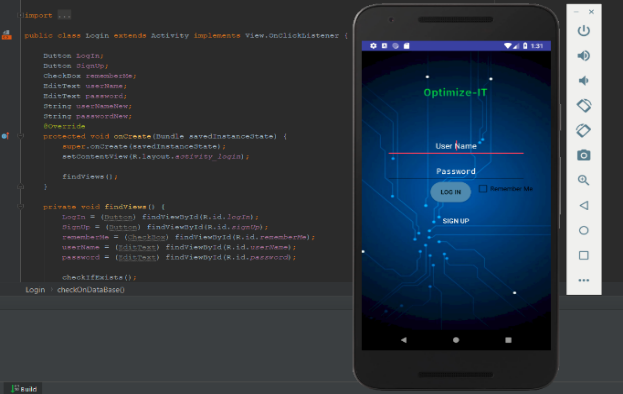
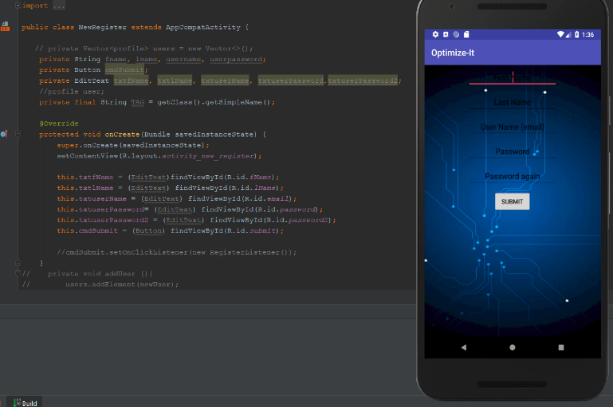
***מבחן 1)***  *=1 , =5*

ישנם 48 שעות שצריך ללמוד בהם למבחן ובגלל שלכל יום הגדרנו 8 שעות למידה, אז צריך ללמוד 6 ימים לפני תאריך המבחן. להתחיל למידה מתאריך23/06/2018 ועד לתאריך 29/06/2018.

***מבחן 2)***  *=0 , =4*

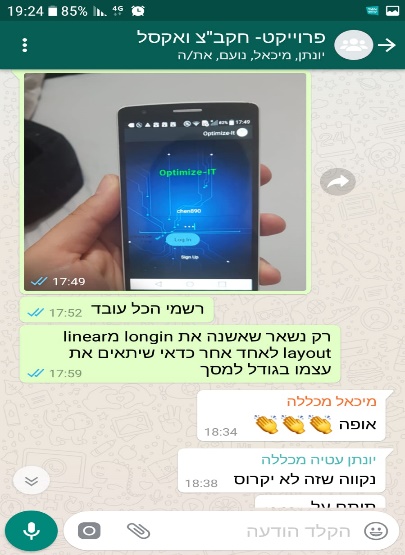
ישנם 32 שעות שצריך ללמוד בהם למבחן ובגלל שלכל יום הגדרנו 8 שעות למידה, אז צריך ללמוד 4 ימים לפני תאריך המבחן. להתחיל למידה מתאריך17/07/2018 ועד לתאריך 20/07/2018.

## **נספח 2: תמונות מקוד התכנות**

******נספח 2.1: כניסה למערכת**  **נספח 2.2: דף רישום משתמש חדש**

## **נספח 3: צילומי מסך – דרכי תקשורת בין חברי הצוות**

### **נספח 3.1: WhatsApp**

****

1. **תסמונת הסטודנט – תופעה בקרב סטודנטים של דחיה מכוונת של משימות** [↑](#footnote-ref-1)
2. **בגרסה זו של המערכת, פרויקט = מבחן**  [↑](#footnote-ref-2)
3. **בגרסה בנוכחית של המערכת, עבודה = למידה** [↑](#footnote-ref-3)
4. **חוק ירקס – דודסון – חוק אמפירי שמתאר את הקשר בין סטימולציה לביצועים.** [↑](#footnote-ref-4)
5. **חוק הופשטטר – "לעולם יידרש לך יותר זמן משציפית" (דאגלס הופשטטר).** [↑](#footnote-ref-5)
6. **נספח 3: צילומי מסך** [↑](#footnote-ref-6)
7. **נספח 1: דוגמה לחישוב ידני** [↑](#footnote-ref-7)