**הצעה לפרויקט גמר** תאריך: 19/2/2023

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **שם התלמיד** | **ת.ז.** | **כתובת** | **עיר** | **טלפון** | **תאריך סיום הלימודים** |
| ברכה אשלג | 213843360 | חזון איש 51 | ירושלים | 058-3246028 | 2023 |
| אילה רודניק | 214092157 | קורדובירו 3 | ירושלים | 058-3241897 | 2023 |

1. **פרטי הסטודנטים:**

שם המכללה: סמינר מרכז בית יעקב

מגמה: הנדסת תוכנה

מסלול: הנדסאים

1. **פרטי המנחה האישי:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **שם המנחה** | **כתובת** | **טלפון נייד** | **תואר** | **מקום עבודה / תפקיד** |
| גב' זהבה מרקס | רח' היעל 15 ירושלים | 052-7127312 | הנדסאי מה"ט | משטרת ישראל |

1. **שם הפרויקט: ניהול ועד בית.**
2. **רקע**
   1. **תיאור ורקע כללי**  
      ניהול ועד הבית הוא פעולה מורכבת הדורשת משאבים רבים כמו: יכולות ארגון, זמן וכו'. ניהול ועד הבית דורש איסוף וניהול כספים, העברת הודעות. המערכת שלנו תאפשר ניהול יעיל, מאורגן ואוטומטי של ועד הבית.
   2. **מטרות המערכת**  
      לאפשר למנהל ועד הבית לתקשר באופן נוח עם הדיירים, לרכז את הגבייה לשמור יומן מאורגן של הוצאות והכנסות.

לאפשר לדיירים תשלום דמי וועד בית באמצעים נוחים וזמינים, שקיפות מלאה של מאזן הקופה, תקשורת נוחה עם מנהל הועד, איתור גורמים אפשריים שפגעו ברכב ששהה בחניון ביתר קלות.

1. **סקירת מצב קיים בשוק**ישנם מספר אתרים המקלים על ניהול ועד בית, אולם אף אחד מהם לא נותן את המענה הנרחב שאתרינו מספק, כמו ניהול החניון הכולל את הרשאות הכניסה והיציאה אליו וממנו ואת החיסכון בסקירה ארוכה של מצלמות האבטחה לאיתור הפגיעות ברכבים בעת השהיה בחניון. וכן אין מערכת אחת המקיפה את כל צרכי ניהול ועד הבית.
2. **מה הפרויקט אמור לחדש או לשפר**הפרויקט יאפשרלכל אדם או חברה המנהלת ועד בית התנהלות פיננסית וארגונית פשוטה וקלה יותר.
3. **דרישות מערכת ופונקציונאליות**
   1. **דרישות מערכת**צד השרת   
      יועמד על שרת IIS המאפשר גלישה אנונימית.

MySql כולל מנגנוני התאוששות מקריסה.  
נפח דיסק פנוי של 30GB וזיכרון מומלץ 16GB.

צד הלקוח   
יוכל לפעול על מכשיר או מחשב אישי המריץ דפדפן google chrome (תיתכן תמיכה בדפדפנים נוספים), רוחב פס 100MBPS לפחות.

* 1. **דרישות פונקציונאליות**המערכת מאפשרת למנהל ועד הבית:
  + להגדיר בניין. (חשבון בנק ורשימת דיירים)
  + להוסיף/ להסיר דירה.
  + להוסיף/ להסיר מודעה ללוח המודעות.
  + לצפות במצלמות השונות בשטח הבניין.
  + לעדכן הרשאות גישה לחניון לדיירים.
  + לעדכן על תשלום דמי ועד בית במזומן.
  + לעדכן על הוצאות (קבועה חודשית או חד פעמית).
  + לחסום לדייר שלא שילם דמי ועד בית את אפשרות הכניסה לחניון.
  + לצפות בהוצאות והכנסות.
  + לפתוח הצבעה לכלל הדיירים בנושאים שונים.
  + להוסיף דייר שלא רוצה להשתמש באפליקציה ולעדכן אותו בהודעות השונות באמצעות אימייל/ (ווצאפ/) SMS.
* המערכת מאפשרת לדייר:
  + לשלם דמי ועד בית באשראי.
  + לשלוח פניה לוועד הבית.
  + לצפות במאזן הוצאות והכנסות ועד הבית.
  + לצפות בקבלות קודמות ולהורידן מהאתר.
  + להשתתף בהצבעה.

1. **בעיות צפויות במהלך הפיתוח ופתרונות**
   1. **תיאור הבעיות**
      1. אם קיים בבניין דייר שלא רוצה להשתמש באתר כיצד נתן לתקשר אתו?
      2. כאשר משתנים הדיירים בדירה מסויימת, כיצד נטפל בכך?
   2. **פתרונות אפשריים**
      1. נשלח לדייר הנ"ל הודעות בדואר אלקטרוני.
      2. בעת עדכון פרטי דירה אם היה שינוי בפרטי הדרים בדירה נמחק את כל הדיירים המתגוררים בדירה זו.
2. **פתרון טכנולוגי נבחר**:
   1. **טופולוגית הפתרון**  
      המערכת תיושם ברשת האינטרנט, ותהיה מורכבת משרת המנהל מסד נתונים יחסי של SQL SERVER, וממשק משתמש בצד הלקוח. התקשורת ביניהם תתבצע באמצעות WEB API.
   2. **טכנולוגיות בשימוש**  
      Node.js   
      השרת יממש Web API כך שקליינטים יוכלו לפנות אליו ולקבל מידע. התשתית שנשתמש בה לצורך כך היא Node.js  
      Node.js היא סביבת ריצה לשפת JavaScript שנכתבה ב-C++ ומבוססת על מנוע V8. היא תוכננה לכתיבת אפליקציות רשת כגון שרתי HTTP.  
      Node.js מאפשרת לקוד JavaScript לרוץ ללא צורך בדפדפן, מה שמאפשר לה לשמש כשפת צד שרת.  
      React   
      React היא ספריית קוד פתוח בשפת JavaScript המשמשת לפיתוח ממשקי משתמש דינמיים, המותאמים לכל סוג מסך ולכל המאפיינים ושינויי המצב של המסך.
   3. **שפות הפיתוח**  
      השרת ייכתב בשפת JavaScript אשר בבסיסה היא שפה שיעודה לצד לקוח , אולם תשתית Node.js מאפשרת להשתמש בה וביתרונותיה גם בצד השרת.  
      חלק נוסף של צד השרת, האחראי על הבינה המלאכותית בפרויקט, ייכתב בשפת Python שהיא נוחה ושימושית מאד לצרכי חישוב ועיבוד.  
      צד הלקוח ייכתב גם הוא בשפת JavaScript (שעליה מבוססת React).  
      בחרנו בשפות אלו שהן פופולריות, שימושיות, ונוחות לפיתוח.
   4. **תיאור הארכיטקטורה הנבחרת**שרת SQL Server יאחסן את הנתונים.

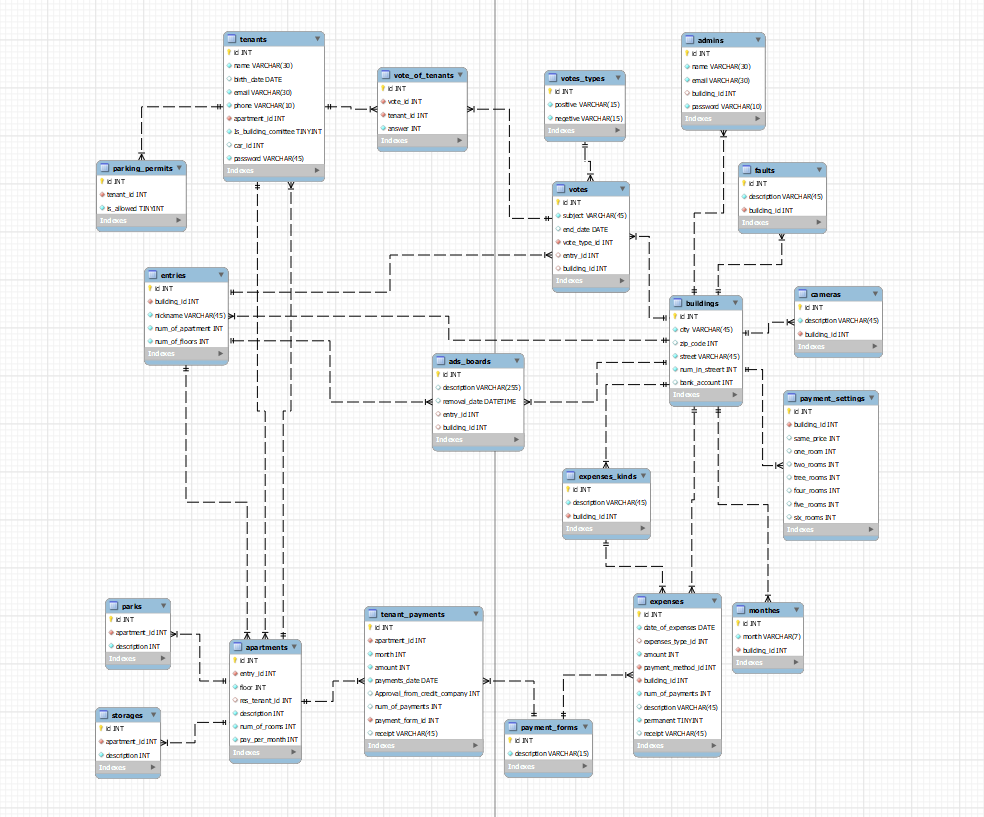
שכבת ניהול הנתונים – Data Layer - תהיה אחראית לשליפת הנתונים הנדרשים ולעדכונם.

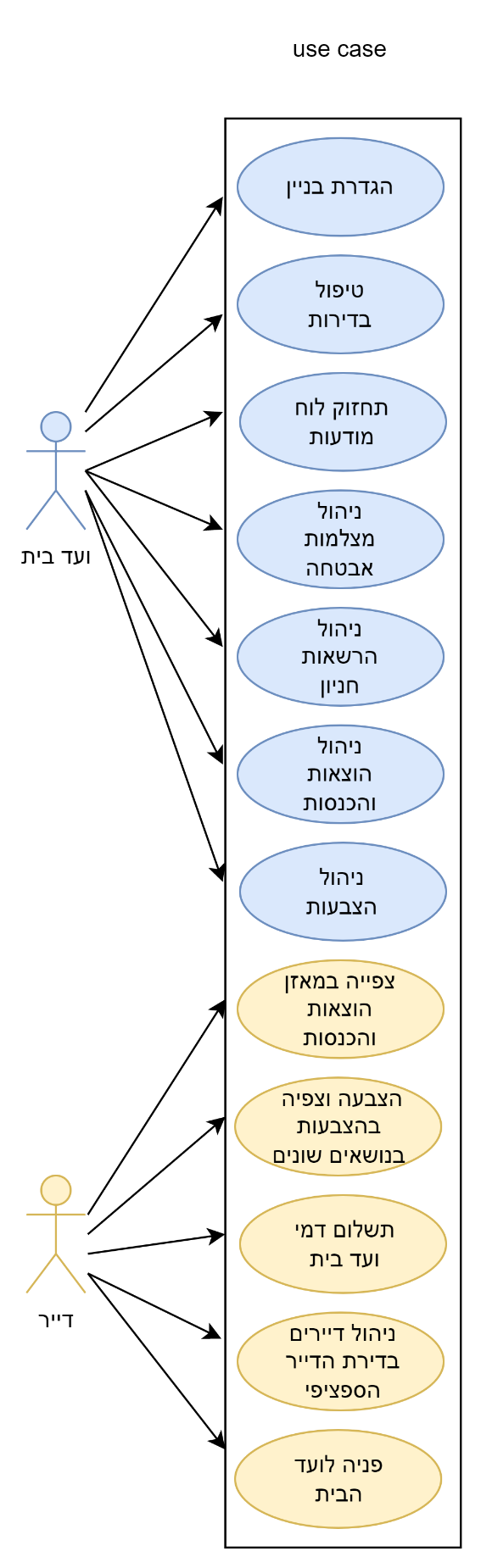
השכבה הלוגית – Business Layer – תנהל את הלוגיקה הנדרשת.

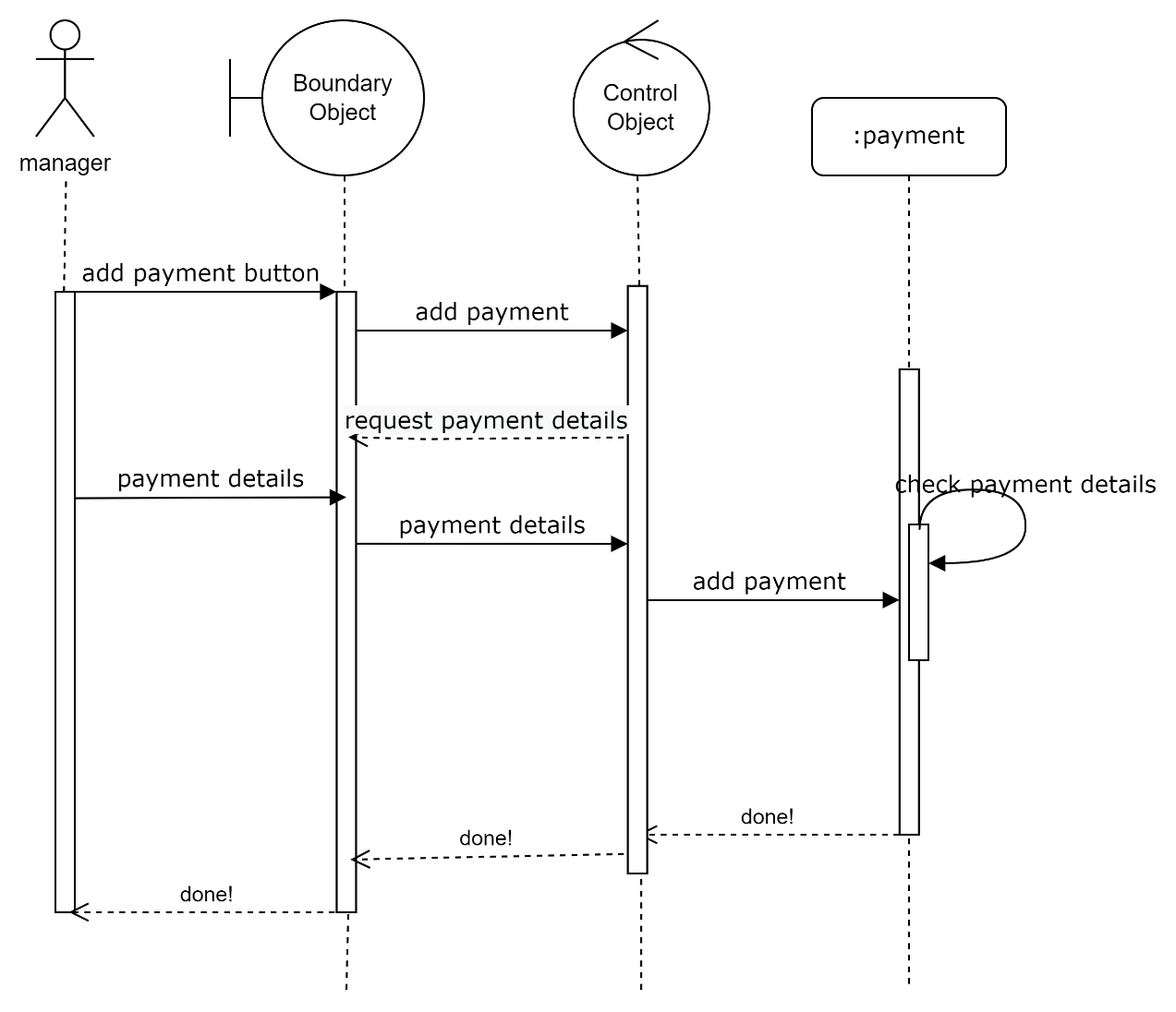
תשתית Node.js תאפשר תקשורת בין השרת ללקוח.

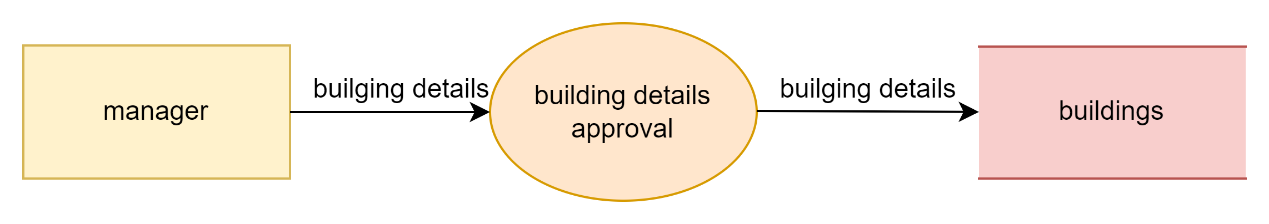
ממשק צד הלקוח ב React יפנה לשרת Node.js לצורך קבלת המידע הנדרש או לצורך עדכונו ויציג את דפי האפליקציה למשתמש.

* 1. **חלוקה לתוכניות ומודולים**חלק א': מכלול תוכנה שרתי, שינהל את מרכיבי הניהול העסקי של המערכת.  
     חלק ב': מכלול הלקוח, שיחשוף ממשק משתמש מעל הweb
  2. **סביבת השרת**שרת IIS יארח את הAPI של צד השרת.
  3. **ממשק המשתמש –GUI**בטכנולוגיית React ניצור דפי אינטרנט שיוצגו על גבי דפדפן chrome (תיתכן תמיכה בדפדפנים נוספים).
  4. **ממשקים למערכות אחרות**לצורך תשלום לוועד הבית נתממשק עם מערכת סליקת אשראי, מערכת שערי החניון וכן עם מערכת מצלמות.
  5. **שימוש בחבילות תוכנה**במהלך הפיתוח נשתמש בNPM ובPIP להתקנת חבילות תוכנה נוספות באם תידרשנה.

1. **שימוש במבני נתונים וארגון קבצים:**
   1. הישויות העיקריות במערכת: דיירים, כניסות, בניינים, דירות, תשלומים  
      
   2. הנתונים יאוחסנו במסד נתונים יחסי של MySQL
   3. לצורך התאוששות מקריסה יופעלו מנגנוני הגיבוי והשחזור של MySQL
2. **תרשימי מערכת מרכזיים**
   1. **Use cases**

****

* 1. **Sequence Diagram  
     **

1. **Data Flow Diagram****תיאור המרכיב האלגוריתמי – חישובי**
   1. הבעיה העיקרית שהפרויקט יפתור היא ניהול יעיל של קופת ועד הבית, על מנת לחסוך כאבי ראש מהמשתמש נציג לו על פי דרישה דו"חות, קבלות, הוצאות, הכנסות או מאזן הוצאות והכנסות.
2. **אבטחת מידע**  
   בנושא אבטחת המידע נסתמך על המנגנונים האוטומטיים של שרת הIIS המארח. אין התייחסות מיוחדת נוספת לנושא אבטחת המידע מכיוון שהאפליקציה אינה עוסקת בתכנים רגישים.

לצורך אבטחת הנתונים הרגישים במערכת נבנה מערכת הרשאות, שתדאג לכך שהמידע הרגיש לא יוצג בפני משתמשים בלתי מורשים. כמו כן נסתמך על תשתית האבטחה של שרת IIS מאובטח.

1. **משאבים הנדרשים לפרויקט**
   1. היקף שעות העבודה כ700 שעות, שיחולקו בין המגישות בצורה מאוזנת.
   2. ציוד נדרש: מחשב PC עם זיכרון RAM של 8GB, תשתית רשת המאפשרת שליחת הודעות דואר אלקטרוני.
   3. תוכנות נדרשות:  
      PyCharm

Visual Studio Code  
MySQL

NodeJS

* 1. ידע חדש שנדרש ללמוד לצורך ביצוע הפרויקט:  
     העמקת הידע וההכרות עם הטכנולוגיות שבשימוש הפרויקט, וכן שימוש במערכת סליקת אשראי, מערכת שערי החניון ומערכת מצלמות.
  2. ספרות ומקורות מידע  
     https://reactjs.org/  
     https://nodejs.org/en/

https://stackoverflow.com

1. **תכנית עבודה ושלבים לביצוע הפרויקט**התכנון והפיתוח החל מחודש אלול תשפ"ב:
   1. תכנון פונקציונאליות המערכת (כולל ציורי מסכים) – כשלושה שבועות.
   2. יצירת תרשימי מערכת- כשבועיים.
   3. תכנון בסיס הנתונים- כשבועיים.
   4. כתיבת הצעת הפרויקט- כשבוע.
   5. פיתוח צד השרת- כעשרה שבועות.
   6. פיתוח צד הלקוח- כעשרה שבועות.
   7. כתיבת ספר הפרויקט- לאורך כל תהליך הפיתוח.
2. **תכנון הבדיקות שיבוצעו**
   1. בדיקות תהליכיות ברמת משתמש – Full Flow

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| מספר בדיקה | מהות הבדיקה | אופן ביצוע הבדיקה | תוצאה רצויה | תוצאה בפועל | עבר או נכשל |
| 1 | הזדהות | הזנת שם וסיסמה, והקשה על אישור | כניסה לדף הבית של האתר (דייר או מנהל) |  |  |
| 2 | הגדרת בניין | הזנת פרטי בניין והקשה על אישור | קבלת הודעה שהמידע נשמר |  |  |
| 3 | הוספת/ הסרת דירה | הזנת פרטי דירה והקשה על אישור | קבלת הודעה שהמידע נשמר |  |  |
| 4 | הוספת/ הסרת מודעה מלוח המודעות | הזנת פרטי מודעה והקשה על אישור | קבלת הודעה שהמידע נשמר |  |  |
| 5 | עדכון הרשאות גישה לחניון לדיירים | הזנת פרטי הדייר ופרטי ההרשאה לשינוי | קבלת הודעה שהמידע נשמר |  |  |
| 6 | עדכון על הוצאה/ הכנסה חדשה | הזנת פרטי ההוצאה/ ההכנסה והקשה על אישור | קבלת הודעה שהמידע נשמר |  |  |
| 7 | הוספת הצבעה | הזנת פרטי ההצבעה והקשה על אישור | קבלת הודעה שהמידע נשמר |  |  |
| 8 | תשלום לוועד הבית באופן ידני | הזנת פרטי התשלום והקשה על אישור | קבלת הודעה שהמידע נשמר והפקת קבלה |  |  |
| 9 | תשלום לוועד הבית באשראי | הזנת פרטי התשלום והקשה על אישור | קבלת הודעה שהמידע נשמר והפקת קבלה עם מס' אסמכתא |  |  |

1. **בקרת גרסאות**לצורך בקרת הגרסאות נשתמש במנגנון הפופולארי Git ובשירותי האחסון של GitHub.

###### \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

###### חתימת הסטודנט חתימת המנחה האישי

1. הערות ראש המגמה במכללה

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **אישור ראש המגמה**

## שם: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ חתימה: \_\_\_\_\_\_\_\_\_\_\_\_ תאריך: \_\_\_\_\_\_\_\_\_\_\_\_\_

1. **הערות הגורם המקצועי מטעם מה"ט**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **אישור הגורם המקצועי מטעם מה"ט**

## שם: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ חתימה: \_\_\_\_\_\_\_\_\_\_\_\_ תאריך: \_\_\_\_\_\_\_\_\_\_\_\_\_