תאריך: 15/01/2020

לכבוד יחידת הפרויקטים מה״ט

**הצעה לפרויקט גמר**

**א. פרטי הסטודנטים**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **שנת סיום הלימודים** | **טלפון נייד** | **כתובת** | **ת.ז.** | **שם הסטודנט** |
| תשפ"א | 052-7651564 | רבן יוחנן בן זכאי 25, אלעד | 332424878 | אליזבת ברינגר |
| תש"פ | 052-716312 | רבן יוחנן בן זכאי 97, אלעד | 201511235 | מירי הורוביץ |

שם המכללה: בית המורה

סמל המכללה: 76076

מסלול ההכשרה: הנדסאים

מגמת לימוד: תכנות מחשבים

מקום ביצוע הפרויקט: בסמינר/בבית/במקום העבודה

**ב. פרטי המנחה האישי**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מקום עבודה/תפקיד | תואר | טלפון נייד | כתובת | שם המנחה \* |
| סמינרים  המכללה למינהל | B.ED  הנדסאי מחשבים | 052-71712954 | חזו"א 11, בני ברק | מרים שמעונוביץ |

חתימת הסטודנט חתימת המנחה האישי חתימת הגורם המקצועי מטעם מה״ט

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



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

1. שם הפרויקט:

kosher list

1. רקע:
   1. **תיאור ורקע כללי:**

האתר kosher list פותח על מנת ליעל את מערכת הכשרות, האתר מאפשר רישום נתונים של משגיחים ועסקים הנתונים תחת ההשגחה -גוף הנמצא תחת השגחתו של משגיח כשרות מקבל תעודת הכשר מהגוף המפקח, והוא רשאי לציין שמוצריו כשרים.

האתר בונה מערכת שעות חודשית לכל משגיח, ובנוסף מאפשר להזין למערכת דוח של ביקור משגיח לתועלת מעקב ואיסוף נתונים.

* 1. **מטרות המערכת:**
* הזנת נתונים של משגיחים ועסקים הנתונים תחת ההשגחה.
* בנית מערכת שעות חודשית לכל משגיח.
* לאפשר למשגיח הזנה של דיווחים על כל ביקור.
* ההנהלה תוכל לעקוב אחר תפקודי המשגיח, והמצב של עסק.
* המערכת תציג לכל משגיח כתובת של ביקור פתע בהתאם לנתונים שהוזנו למערכת.

1. סקירת מצב קיים

לא מצאנו אתר מעין זה.

(המערכת מתנהלת באמצעות מזכיר שדואג לעדכן את המשגיחים טלפונית או באמצעות מיילים).

1. מה הפרויקט אמור לחדש או לשפר?

רצינו בעצם להקל על המערכת, על מנת לחסוך במשאבים ולדייק את התוצאות ולכן האתר אוסף מידע מהמשגיחים, על סמך דוחו"ת דיווח- ועל סמך זה בונה מערכת שעות אוטומטית, ומתחשב בכל האילוצים המדווחים והמתבקשים.

1. דרישות מערכת ופונקציונאליות
   1. דרישות מערכת, סביבת הטמעה ושימוש.

המערכת תעבור קומפילציה והפצה בסביבת visual studio עם התקנת Net Framework

גירסה 0.4 ומעלה והיא אמורה לרוץ בסביבת שרת אשר מריץ Express IIS לקבלת בקשות

לתצוגת דפי אינטרנט.

המשתמש יוכל להריץ את האתר בכל מכשיר אשר מותקן עליו דפדפן אינטרנט.

* 1. שרידות, ביצועים / התמודדות עם עומסים:

צד השרת מריץ IISExpress המסוגל להתמודד עם מספר קריאות של קריאות בו זמנית.

גם עומס על שרת ה-SQL אינו צפוי בסדר גודל כזה של אתר מכיוון שהוא בנוי להתמודדות

בהצלחה עם עומסים כבדים בהרבה.

* 1. דרישות פונקציונאליות

1. בעת הרשמת משתמש למערכת, עליו להזין את פרטיו, שיקלטו ויוזנו למאגר.
   1. משתמש יוכל להתחבר למערכת כמנהל או כמשגיח, באמצעות הזנת סיסמה ושם משתמש.
2. האתר מחשב את המרחקים הגיאוגרפיים בין המשתמש לבין העסק לצורך בניית מערכת השעות לעובדים.
3. תהיינה הרשאות צפיה שונות ומשתנות על פי סוג המשתמש ותפקידו
   1. **משתמש משגיח:**
      1. יוכל לצפות ולהתעדכן במערכת השעות החודשית שלו.
      2. יוכל להזין למערכת דו"ח על ביצוע משימה, שעות ביקור, תוצאות הביקור, הערות נצרכות.
   2. **משתמש מנהל:**
      1. מתן אישור לקליטת עובד חדש למערכת
      2. מנהל יצפה במערכת של כלל העובדים, ויוכל לערוך ולתזמן עבור כל משגיח את שעות העבודה והעסקים שהוא יקבל.
4. **דירוג וטיפול בעסקים בדירוג גבוה:**
   1. האתר מחשב את התוצאות הביקור שהזין המשתמש בביקור סדיר בעסק, ועפ"י תוצאות אלו בונה מאגר של עסקים בסיכון, במידה ויישנם, ומסנן אותם עפ"י רמת הדחיפות לביקור משגיח.
   2. על מנת לממש ביקורי פתע בעסקים שהמערכת הגדירה כ"עסקים בסיכון מוגבר": המערכת תציג לכל משתמש שייכנס למאגר הנ"ל את העסק בסיכון הכי מוגבר.
5. המשתמש יוכל להזין נתונים גאוגרפיים, ע"מ שהמערכת תתחשב גם בהם בעת התאמת ה"עסק בסיכון" שבו עליו לבקר.
6. בעיות צפויות במהלך הפיתוח ופתרונות
   1. תיאור הבעיות:
      * **בעיה 1:** המערכת בונה עבור כל עובד מערכת שעות חודשית, מאחר וישנם משתמשים רבים, יקשה על משתמש למצוא את השעות שהמערכת תזמנה עבורו, מתוך כלל הנתונים.
      * **בעיה 2:** כיצד נזהה משתמש שמעונין להיכנס למערכת.
      * **בעיה 3:** ישנה אפשרות של המערכת להתחשב בנתונים גאוגרפיים ע"מ לבחור למשתמש את העסק הקרוב אליו ביותר, כדי שהמערכת תזהה את העיר- על המשתמש להקליד אותה בדייקנות כפי שהיא מופיעה במאגר.
      * **בעיה 4 :** יש צורך להחליט על מדד על פיו נכריז על "עסקים בסיכון גבוהה" שנידרשים לביקורים תכופים יותר
   2. פתרונות אפשריים:
      * **לבעיה 1:**

**פתרון 1:** כל משתמש מתחבר עם שם משתמש וסיסמה פרטיים ויוכל לצפות רק במערכת השעות של אותו משתמש. מלבד מי שיתחבר כמנהל ויהיה רשאי לצפות בכל הנתונים של כלל המשתמשים.

**פתרון 2:** המערכת תציג אפשרות של חיפוש טבלאות לפי משגיח

**פתרון 3 :** כל שם משתמש יודגש בצבע אחר בטבלה ובכך יקל על ההמצאות

* + - **לבעיה 2:**

**פתרון 1:** לפי שם משתמש ות.ז

**פתרון 2 :** לפי מייל

* + - **לבעיה 3:**

**פתרון 1:** נתכנן את המערכת שתנסה לחפש את העיר שהכי דומה לנתונים שהוקשו.

**פתרון 2:** נציג למשתמש רשימה נגללת עם שמות של כל הערים שקיימים במערכת- לבחירה.

* + - **לבעיה 4:**

**פתרון 1:** חישובים סטטיסטיים שינתחו את דיווחי המשגיחים על העסקים, ידרגו את מדד העסקים.

**פתרון 2:** הכרזה על מדד שרירותי, על פיו עסק שיזכה במספר תלונות מסוים. ידורג בהתאם.

**פתרון 3:** החלטת מנהל.

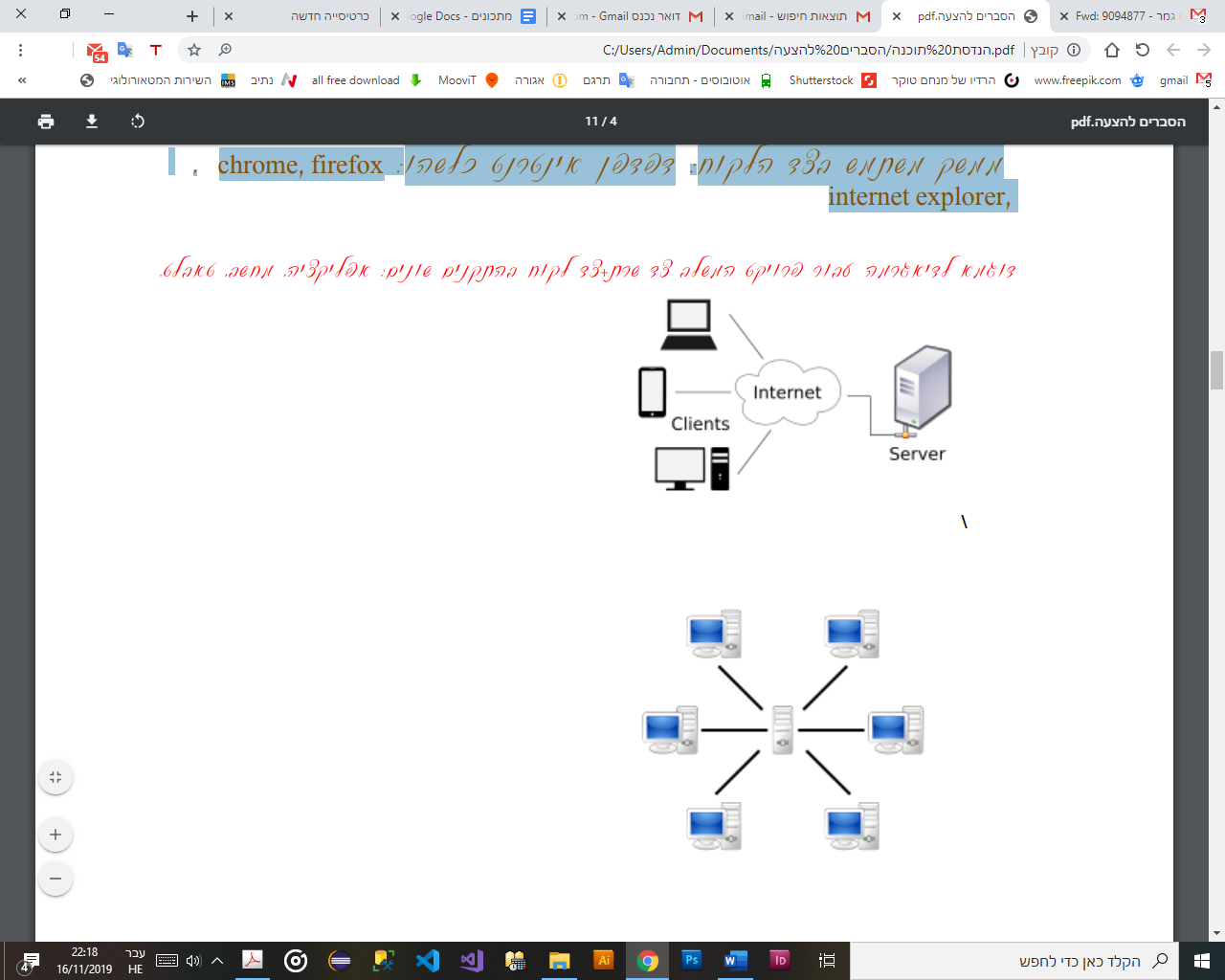
## **הפתרונות הנבחרים:**

* + - לבעיה 1: כל משתמש מתחבר עם שם משתמש וסיסמה פרטיים ויוכל לצפות רק במערכת השעות של אותו משתמש. מלבד מי שיתחבר כמנהל ויהיה רשאי לצפות בכל הנתונים של כלל המשתמשים.
    - לבעיה 2: לפי שם משתמש ות.ז
    - לבעיה 3: נציג למשתמש רשימה נגללת עם שמות של כל הערים שקיימים במערכת- לבחירה.

1. פתרון טכנולוגי נבחר:

## טופולוגית הפתרון

המערכת מורכבת משרת IIS המריץ את האתר בסביבת ה- server, sql-server של DB’s - נתונים מסד ממשק משתמש בצד הלקוח: דפדפן אינטרנט כלשהו: firefox, chrome , internet explorer,



## טכנולוגיות בשימוש.

Entity Framework

client - Angular

## שפות הפיתוח

* שפות הפיתוח בצד השרת: c#
* שפות הפיתוח בצד הלקוח: Type script, html5, css3

## ארכיטקטורה נבחרת

נשתמש במודל השכבות של Entity framework נשתמש בשיטה זו כדי לקיים הפרדה בין השכבות השונות בפרויקט (DAL, BL, GUI). היתרון הכללי בחלוקה לשכבות הוא תחזוקה קלה של המערכת וכך גם ניתן לטפל בכל שכבה בנפרד בלי לגעת בשכבות האחרות, תורם לעבודה יעילה ומסודרת.

## חלוקה לתכניות ומודולים.

1.שכבת תצוגה ( Presentation Layer) שכבת ממשק המשתמש .(UI- User Interface) שכבת התצוגה מדפי HTML באתר אינטרנט השכבה מתקשרת בין המשתמש לבין שאר שכבות המערכת.

2.שכבת הלוגיקה העסקית – (BL - Business Logic) השכבה שאמונה על הלוגיקה של המערכת, עוסקת בעיבוד מידע, בחישובים שונים ושליחתו לשכבת התצוגה. בשכבה זו נממש את הפונקציונליות של המערכת.

3.שכבת הנתונים - (Data Access Layer - DAL) שכבה זו מורכבת ממקור נתונים- מסד הנתונים שלנו וממערכת תוכנה Entity Framework אשר תפקידה לקרוא את המידע הנדרש למערכת, לשמור את העדכונים, ולהוסיף מידע חדש.

סביבת השרת   
לצורך הפרויקט נשתמש בשרת מקומי- IIS Express המסופק עם סביבת העבודה של Visual Studio. אם האתר ירכש על ידי לקוח נעלה אותו לשרת אירוח כלשהו או- Microsoft Azure

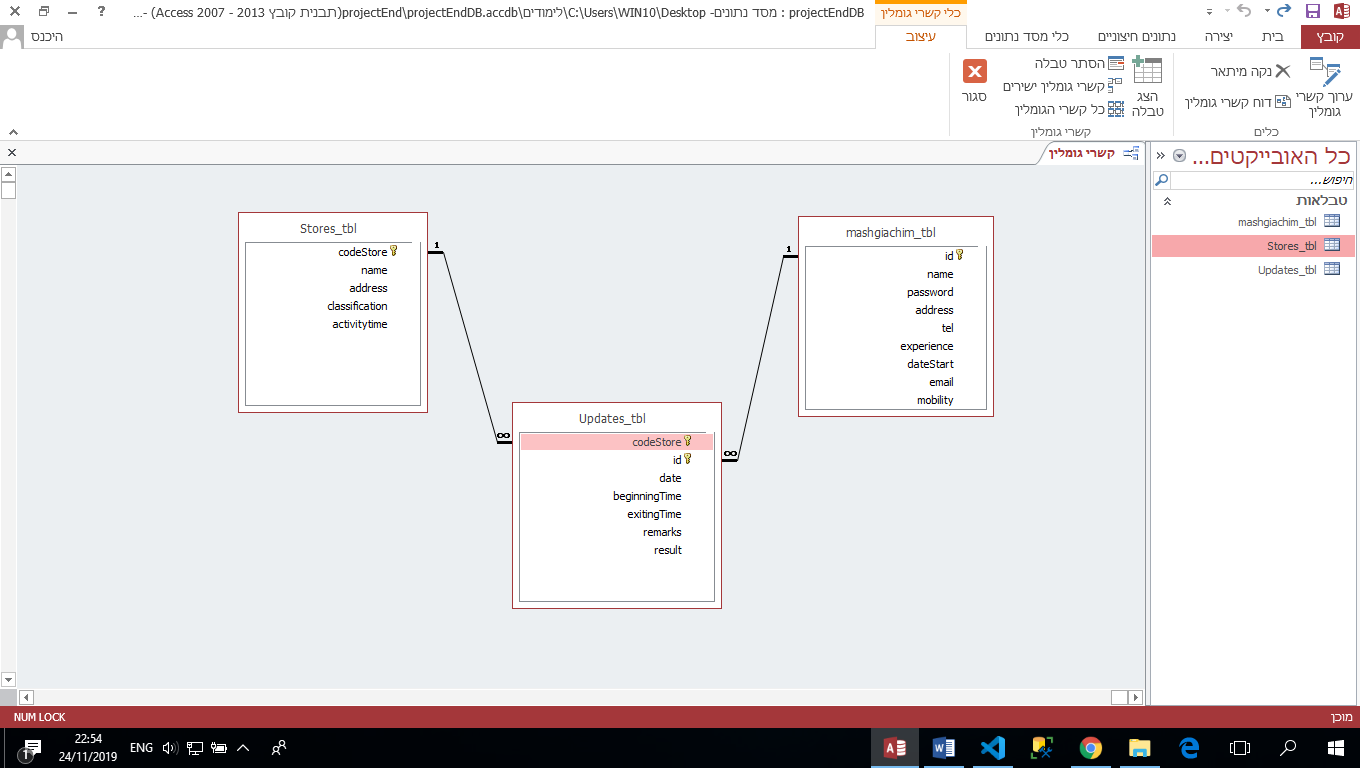
## ממשק המשתמש/לקוח – GUI שכבת ה-GUI מורכבת מדפי HTML שמוצגים למשתמש דרך הדפדפן.

## ממשקים למערכות אחרות / API: - לא קימים

## שימוש בחבילות תוכנה Bootstrap, Entity Framework, Angular Material

1. **שימוש במבני נתונים וארגון קבצים**

## מבני הנתונים



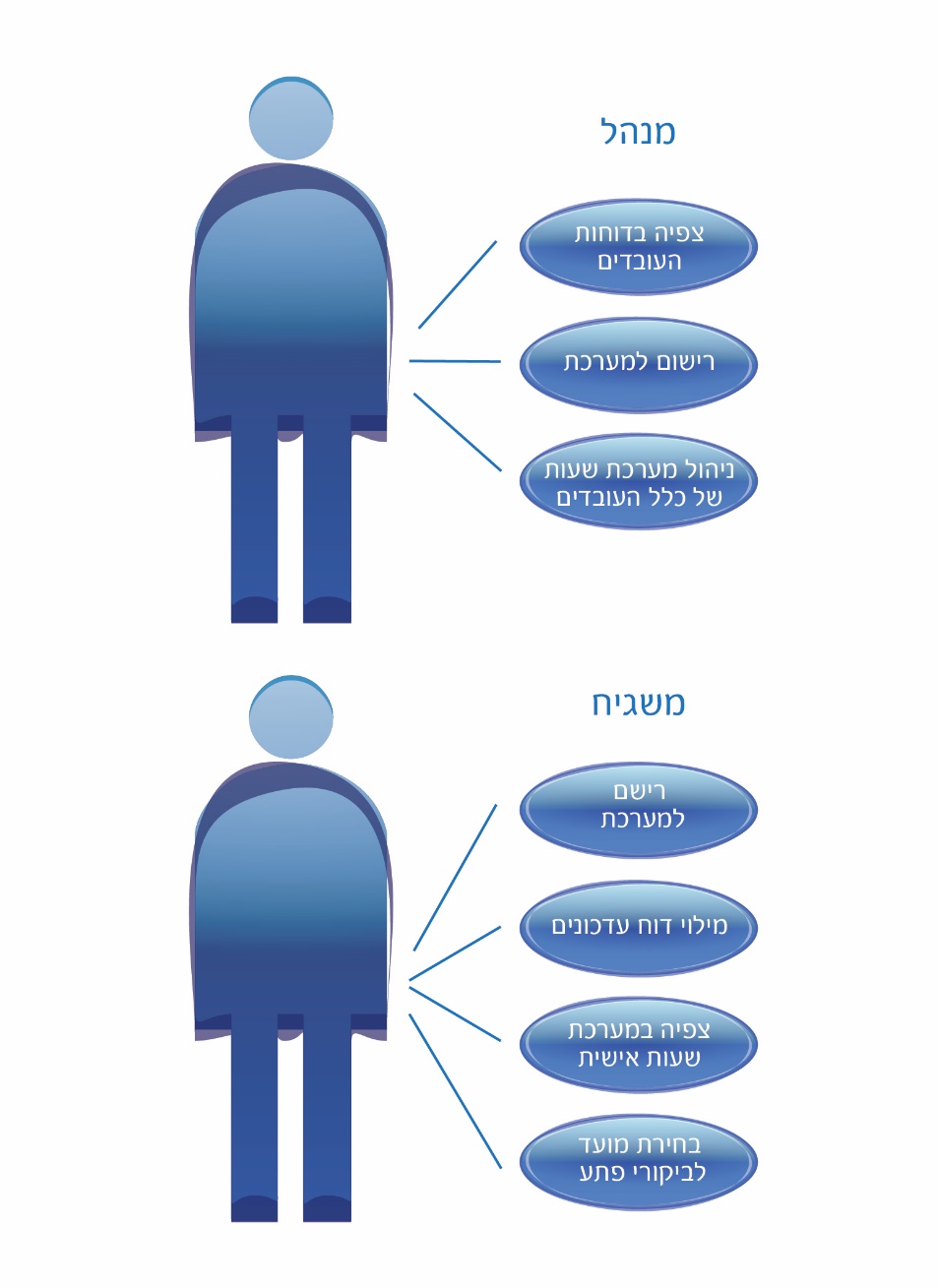
## שיטת האחסון שיטת האחסון היא על ידי מסד נתונים- SQL Server הגישה למסד נתונים היא על ידי Entities

## מנגנוני התאוששות מנפילה/ קריסה/ תמיכה בטראנזקציות.

## SQL Server- בנוי לתמיכה במס' גדול של קריאות בו זמניות ואין חשש לקריסה ברמה של הפרויקט.

1. **תרשימי** **מערכת** **מרכזיים**

## Use Case



1. תיאור המרכיב האלגוריתמי – חישובי.

## איזה בעיה בא לפתור, איך יפתור?

כאשר משגיח נכנס לדף "ביקורי פתע" ורוצה לקבל רשימה של חנויות אפשרויות. נרצה להציג לו את החנות שהכי דחוף עכשיו למערכת שהוא יבקר בה.

כדי שנוכל לדעת מה החנות שהכי חשוב לבקר בה, נאסוף נתונים על החנות וננתח אותם.

## איסוף מידע וניתוחים סטטיסטיים (אנליטיקות)

המערכת תאסוף נתונים מדו"חות משגיחים - תנתח ותדרג אותם, על מנת להסיק מהי החנות שהנתונים שהתקבלו עליה הם הבעייתיים ביותר.

1. תיאור/התייחסות לנושאי אבטחת מידע

על מנת לשמור על רמת אבטחה מידע גבוהה, כל אדם אשר ירשם במערכת יצור לעצמו שם משתמש וסיסמה, שיקנו לו גישה לפעולות נוספות במערכת.

על מנת ליצור מידור ביו המשתמשים, משתמשים רגילים לא יוכלו לבצע שינוים ולצפות במידה של משתמשים נוספים. מלבד משתמש מנהל שמקבל הרשאות מיוחדות.

1. משאבים הנדרשים לפרויקט:

## מספר שעות המוקדש לפרויקט:720

## חלוקת עבודה בין חברי הצוות: עבודה משותפת

## ציוד נדרש: מחשב הכולל: חיבור לאינטרנט , RAM, 8GB HD SSD, i5 CPU

## תוכנות נדרשות: Visual Studio, Visual Studio Code, Sql server

## ידע חדש שנדרש ללמוד לצורך ביצוע הפרויקט: איך להשתמש ב- Angular-Material.

## ספרות ומקורות מידע: stackoverflow.com, w3schools.com, material.angular.io

1. תכנית עבודה ושלבים למימוש הפרויקט:

|  |  |
| --- | --- |
| שלב | החל מתאריך |
| ייזום | אוגוסט |
| ניתוח מבני נתונים | ספטמבר |
| איפיון UI | אוקטובר |
| כתיבת לוגיקה עסקית | 01.8.20-01.11.20 |
| עיצוב | 01.8.20-01.11.20 |
| QA | דצמבר |
| התקנה הטמעה והדרכה | פברואר |

1. תכנון הבדיקות שיבוצעו

## בדיקות תהליכיות ברמת משתמש בהן נדרשת המערכת לעמוד .( Full Flow)

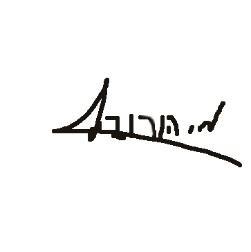
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| מספר בדיקה | מס` דרישה במסמך אפיון | מקרי הבדיקה | ידנית/ אוטומטית | חשיבות | הערות |
| 1 | 5.3 | תקינות מייל | ידנית | גבוהה |  |
| 2 | 5.3 | תקינות סיסמא | ידנית | גבוהה |  |
| 3 | 5.3 | בדיקת הרשאות | ידנית | גבוהה |  |
| 4 | 5.3 | בדיקה שלעובד יש מערכת שעות מלאה | ידנית | בינונית |  |
| 5 | 5.3 | בדיקה שתוצאות החישוב מציגות את העסק בסיכון המוגבר ביותר | ידנית | גבוהה |  |

1. בדיקות יחידה (unit test)

לא יתבצעו בדיקות יחידה אוטומטיות

1. בקרת גרסאות (version control)

נשתמש ב it HubG על מנת לשמור גרסאות ולשתף בקוד. כמו כן נוכל לדאוג לגיבוי בעת הצורך.



**חתימת הסטודנט חתימת הסטודנט חתימת המנחה**

**\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_**

הערות ראש המגמה במכללה:

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

אישור ראש המגמה:

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

הערות הגורם המקצועי מטעם מה"ט:

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

אישור הגורם המקצועי מטעם מה"ט:

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