

תאריך: 03/10/2018

לכבוד

יחידת הפרויקטים מה"ט

**הצעה לפרויקט גמר**

**א. פרטי הסטודנטים**

שם הסטודנט	ת.ז. 9 ספרות	כתובת	טלפון נייד	תאריך סיום הלימודים	שם המכללה:
תמי יוסקוביץ	319091435	פדרמן 9 קרית הרצוג- בני ברק	0533104271	מרץ 2019	בית המורה – סמינר שצ'רנסקי
נעמי פישביין	207145244	רשב"ם 22 בני ברק	0504112152	מרץ 2019	

סמל המכללה: 76076

מסלול ההכשרה: הנדסאים

מקום ביצוע הפרויקט: בסמינר

**ב. פרטי המנחה האישיים:**

שם המנחה *	כתובת	טלפון נייד	תואר	מקום עבודה/תפקיד
מרים שמעונוביץ	חזון איש 1	052-717195	תואר הנדסאי +B.ED	מרצה בסמינר + המכללה למנהל

חתימת הגורם המקצועי מטעם מה"ט

חתימת המנחה האישי

חתימת הסטודנט

## **1. רקע:**

### **1.1 תיאור ורקע כללי:**

האתר comfiler מציע ניהול קבצים באופן חכם ובטוח, מותאם לכל ארגון או מערכת באשר הוא. האתר ינהל את שמירת הקבצים בשרת, והגישה אליהם תהיה רק באמצעות האתר. האתר ידאג לשמור מידע נצרך על הקבצים ויספק מנגנון חיפוש מורחב עליהם. פתיחת קובץ תתאפשר רק למשתמש בעל הרשאת גישה ותגרום להורדה של הקובץ למחשב האישי. לאחר הצפייה/העריכה-בעת סגירת הקובץ, יישמרו השינויים בשרת באופן אוטומטי והקובץ יימחק מתקני ההורדות של המשתמש.

### **1.2 מטרות המערכת:**

- המטרה העיקרית היא ניהול קבצים באופן חכם ויעיל, שיהיה קל לאתרם גם לאחר זמן רב.
- אבטחת הקבצים מפני אובדן, כגון מחיקה ללא שימת לב, שכחה אנושית של מיקום הקובץ. וכדו' שמירת הקבצים בשרת אחד. וגישה אליהם רק באמצעות האתר ורק עם הרשאת גישה מתאימה, וע"י כך מניעת העתקה מרובה של הקבצים למחשבים אישיים. דבר ששומר על הקניין הרוחני של הקבצים.
- הערה: המערכת אינה מתימרת לחסום כל אפשרות של זליגת מידע החוצה. בסופו של דבר חברי הארגון הם אלה שמיצרים את המידע השמור בקבצים ואם ברצונם להפיצו, לא ניתן למנוע זאת מהם במסגרת מערכת ממוחשבת. (שהרי באפשרותם פשוט להקליד פעמיים את תוכן הקובץ)
- המערכת תפותח באופן שימנע זליגת מידע שנעשית בטעות ובלי שימת לב, וכן תקשה על העתקת הקובץ, בין השאר ע"י הרשאות גישה, מחיקת הקובץ לאחר הסגירה וכו'.

## **2. סקירת מצב קיים בשוק, אילו בעיות קימות:**

בארגונים רבים הקבצים נשמרים ע"פ החלטה רגעית של יוצר הקובץ. במהלך השנים נשכח מיקומם או שמם של הקבצים, אין מידע מסודר על ההיסטוריה שלהם וכך עלולים לאבד חומר חשוב. כמו"כ לעיתים הארגון מעוניין לאפשר גישה לעובד מסוים לקובץ מסוים לצורך ספציפי. אך אינו מעוניין להשאירו ברשותו כעותק לזמן בלתי מוגבל, דבר העלול לגרום לפגיעה בקניין הרוחני של קבצי הארגון.

## **3. מה הפרויקט אמור לחדש או לשפר:**

- ניהול של שמירת הקבצים בשרת וגישה אליהם רק באמצעות האתר.
- ניתן יהיה לצפות בקובץ/ לערוך אותו רק עם הרשאת גישה מתאימה.
- שמירת פרטי מרובים על הקובץ, כולל שינויים ועידכונים הנעשים בו במהלך חייו.

- יכולת חיפוש מורחבת ונוחה, המאפשרת סינון ע"פ פרמטרים רבים.
- בכדי למנוע עותקים מרובים של הקובץ במחשבים אישיים- המערכת תדאג לשמירה אוטומטית של הקבצים בשרת, תוך מחיקתם מהמחשב האישי.

#### **4. דרישות מערכת ופונקציונאליות:**

##### **4.1 דרישות מערכת:**

- סביבת הטמעה ושימוש. שרידות, ביצועים\התמודדות עם עומסים.
1. המערכת תעבור קומפילציה והפצה בסביבת visual studio עם התקנת . Net Framework גרסה 4.0 ומעלה והיא אמורה לרוץ בסביבת שרת אשר מריץ IIS Express לקבלת בקשות לתצוגת דפי אינטרנט.
  2. על מנת לייעל את מנוע החיפוש, ולקצר את זמן שליפת הנתונים ממסד הנתונים, הנתונים יישמרו בתקיות בארכיון קבועה של חלוקה לפי תאריך (שנה וחודש), וכך יוקל על המחשב לשלוף את הנתונים המבוקשים במהירות המקסימלית.

##### **4.2 דרישות פונקציונאליות:**

#### **רשימת דרישות המשתמש מהמערכת, מהן הפעולות בהן נדרשת המערכת לתמוך.**

- 4.2.1 המשתמש מקיש שם וסיסמה בדף ה-Login. המערכת מבצעת בדיקה האם המשתמש קיים.
- 4.2.2 האתר יכלול מערכת סינון וחיפוש קבצים ע"פ קריטריונים כגון:  
שם הקובץ, שם יוצר הקובץ, תאריך יצירתו, תאריך עדכון, קטגוריה וכו'....
- 4.2.3 המערכת תציע אפשרות של יצירת קובץ.
- 4.2.4 המערכת תציע אפשרות של יצירת תבנית.
- 4.2.5 המשתמש יבצע סינון על הקבצים ע"פ רצונו.
- 4.2.6 המשתמש בוחר קובץ להורדה ומתבצעת הורדה של הקובץ הנוכחי לתקיית ה"הורדות" במחשב המשתמש.

4.2.7 המערכת תאפשר פתיחה של הקובץ הנבחר ע"י המשתמש, ע"פ הרשאת הגישה שקיימת למשתמש זה לקובץ זה כגון: קריאה בלבד/ עריכה.

4.2.8 המשתמש בוחר קובץ ומוריד אותו למחשב האישי, מתבצעת בדיקה ע"י המערכת האם הקובץ פתוח אצל משתמש אחר. אם הקובץ אכן נמצא פתוח, יפתח הקובץ לקריאה בלבד וייתן על כך פידבק למשתמש

4.2.9 במידה והקובץ הנבחר לא בשימוש, המערכת תבדוק האם יש למשתמש הרשאת עריכה לקובץ. אם כן- הקובץ יפתח לעריכה, אם לא- ייתן על כך פידבק למשתמש.

4.2.10 כאשר המשתמש בוחר ביציאה מהקובץ, מתבצעת העלאה של הקובץ לשרת אם המשתמש הוא בעל הרשאת עריכה המערכת תציג לו הודעה האם מעוניין לשמור את הקובץ בגירסה חדשה, וע"פ בחירתו המערכת תשמור את הקובץ בשרת. אם משתמש אינו בעל הרשאת עריכה, המערכת לא תשמור גרסה נוספת לאותו קובץ ותבצע רק מחיקה של הקובץ מתיקית ההורדות.

## **5. בעיות צפויות במהלך הפיתוח ופתרונות (תפעוליות, טבנולוגיות, עומס ועוד):**

### **5.1 תיאור הבעיות- הללו כפועל יוצא של דרישות המשתמש מהתוכנה:**

1. בפתיחה ושימוש בקובץ מסוים ע"י שני משתמשים ויותר באותה עת, עלול להיווצר קונפליקט של גרסאות בין הקבצים- לדוג' משתמש א ומשתמש ב פותחים קובץ א. משתמש א' מוחק את שורה 23, באותה עת משתמש ב' עורך את שורה 23 בלי ידיעה שהיא נמחקה כבר ע"י משתמש אחר. כאשר שני המשתמשים ינסו לשמור את השינויים שערכו יישמר רק השינוי האחרון.
2. תוכנות האופיס אינן יודעות להכיר בין הקבצים המיוחדים השייכים למערכת, לבין קבצים רגילים הקיימים במחשב הלקוח, ועל כן אינן יודעות מתי להוסיף את התוסף המיוחד השייך לקבצי המערכת.

### **5.2 פתרונות אפשריים. (נא ציין פתרונות אפשריים וחלופות ארכיטקטוניות):**

1. בפתיחת קובץ ע"י משתמש הקובץ יינעל, ולא תהיה אפשרות לעריכה ע"י משתמש אחר, המשתמש יקבל פידבק מרמת ה"ממשק משתמש" שהקובץ נעול, כיוון שהוא נמצא בשימוש ע"י משתמש אחר.
2. לצורך פתרון בעיה זו ניתן שם מיוחד לכל קובץ מערכת ע"י שימוש במחלקת Guide היוצרת קוד יוניקי המורכת מ 32 ספרות, וכך תוכנות האופיס יוכלו לזהות בקלות את הקבצים המיוחדים של המערכת, ולהוסיף להם את התוסף.

## 6. פתרון טכנולוגי נבחר:

### 6.1 טופולוגית הפתרון- כלומר: פרישת המערכת , היכן יתבצע יישום המערכת

(deployment), מרכיבי הפרישה. הנ"ל ברמת מערכת (לדוג' פרויקט פיתוח אתר אינטרנט: המערכת מורכבת משרת, ממשק משתמש בצד הלקוח, DB's, טווח תקשורת-אינטרנט, המערכת תיושם ברשת האינטרנט , יש להציג את דיאגרמת המערכת וכף):

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

### 6.2 טכנולוגיות בשימוש.(איזה ומדוע בכמה מילים)

**Asp .Net MVC Razor** - הפרויקט הוא אתר אינטרנט. טכנולוגית MVC מותאמת לצרכים ולדרישות של האתר מבחינת הסדר והאירגון הלוגיסטי של המערכת.  
**Jquery** - טכנולוגיה המאפשרת עיצובים ואנימציות עדכניים, על מנת לספק למשתמש חווית גלישה מרמה מקסימלית.

### 6.3 שפות הפיתוח:(איזה שפות ומדוע בכמה מילים?)

**Javascript** - שפה זו היא שפה בסיסית ופופולרית המשמשת לפיתוח אפליקציות web, משמשת לכתיבת קוד פונקציונלי, על מנת לאפשר דינמיקה בצד הלקוח.  
**C#** - שפה זו היא שפת תכנות פופולארית שמיועדת לפיתוח כללי של מגוון אפליקציות בכל התחומים. התחביר והעקרונות שלה פשוטים מצד אחד, אך עשירים ביכולות מצד שני. משמשת לכתיבת קוד פונקציונלי של שליפות נתונים ממסד הנתונים, וכן חישובים מורכבים השייכים לצד שרת.

### 6.4 תיאור הארכיטקטורה הנבחרת- הסבר בכמה מילים מדוע

הארכיטקטורה הנבחרת היא חלוקה ל-3 שכבות Architecture Tier. 5-בפרויקטים שמבוססים על Database מומלץ להשתמש בגישה של מודל השכבות , בארכיטקטורה זו קיימת הפרדה בין השכבות השונות בפרויקט. היתרון העיקרי בחלוקה לשכבות הוא תחזוקה קלה, במיוחד במערכות גדולות, כך שניתן יחסית בקלות להחליף כל שכבה בלי לגעת בשכבות האחרות כאשר נרצה לעשות שינויים בתוכנית.

### 6.5 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 השכבות של המערכת.
2. שכבת הלוגיקה העסקית (BL - Business Logic) – השכבה שאמונה על הלוגיקה של

המערכת, עוסקת בעיבוד המידע, בחישובים שונים ושליחתו לשכבת התצוגה.

בשכבה זו נממש את הפונקציונאליות של המערכת.

3. שכבת הנתונים (Data Access Layer - DAL) - שכבה זו מורכבת ממקור נתונים – מסד

הנתונים, וממערכת תוכנה entity framework אשר תפקידה לקרוא את המידע הנדרש

למערכת, לשמור את העדכונים, ולהוסיף מידע חדש או למחוק פרטי מידע קיימים.

### **6.6 סביבת השרת (מקומי, וירטואלי, ענן, שירות אירוח) :**

לצורך הפרויקט נעשה שימוש בשרת מקומי – IISExpress , המסופק עם סביבת העבודה של visual studio.

### **6.7 ממשק המשתמש/לקוח – GUI :**

שכבת ה GUI מורכבת מדפי HTML שמוצגים למשתמש דרך הדפדפן.

### **6.8 ממשקים למערכות אחרות / API :**

אין

### **6.9 שימוש בחבילות תוכנה:**

JQuery, Bootstrap, CSS, Entity Framework

## **7. שימוש במבני נתונים וארגון קבצים:**

### **7.1 מבני הנתונים**

**קבצים:** קוד , תיאור , קוד יוצר קובץ , תאריך יצירת קובץ , קוד מעדכן קובץ , תאריך עדכון אחרון , סיומת , גירסא.

**פרטי קבצים:** קוד יוצר קובץ , קטגוריה , תיאור , הערה.

**קטגוריות:** קוד קטגוריה , תיאור.

**סיומות של קבצים:** קוד סיומת , תיאור.

**היסטוריה:** קוד , מספר גירסא , קוד יוצר גירסא , תאריך עדכון הגירסא , הערה.

**משתמשים:** ת"ז , סיסמא , אימייל , שם משתמש.

**שיתוף:** קוד קובץ , ת"ז משתמש , רמת שיתוף.

**רמות שיתוף:** קוד , תיאור.

**הרשאות:** קוד , תיאור.

**קשר בין הרשאות לתפקידים:** קוד הרשאה , קוד תפקיד.

**תפקידים:** קוד , תיאור.

**קשר בין משתמשים לתפקידים:** קוד תפקיד , ת"ז משתמש.

**תבניות:** קוד , תיאור

### **7.2 שיטת האיחסון (מאגר, קבצים ובאיזה טכנולוגיה)**

שיטת האחסון היא ע"י Sql Server - מסד נתונים. הגישה למסד הנתונים היא ע"י Entities.

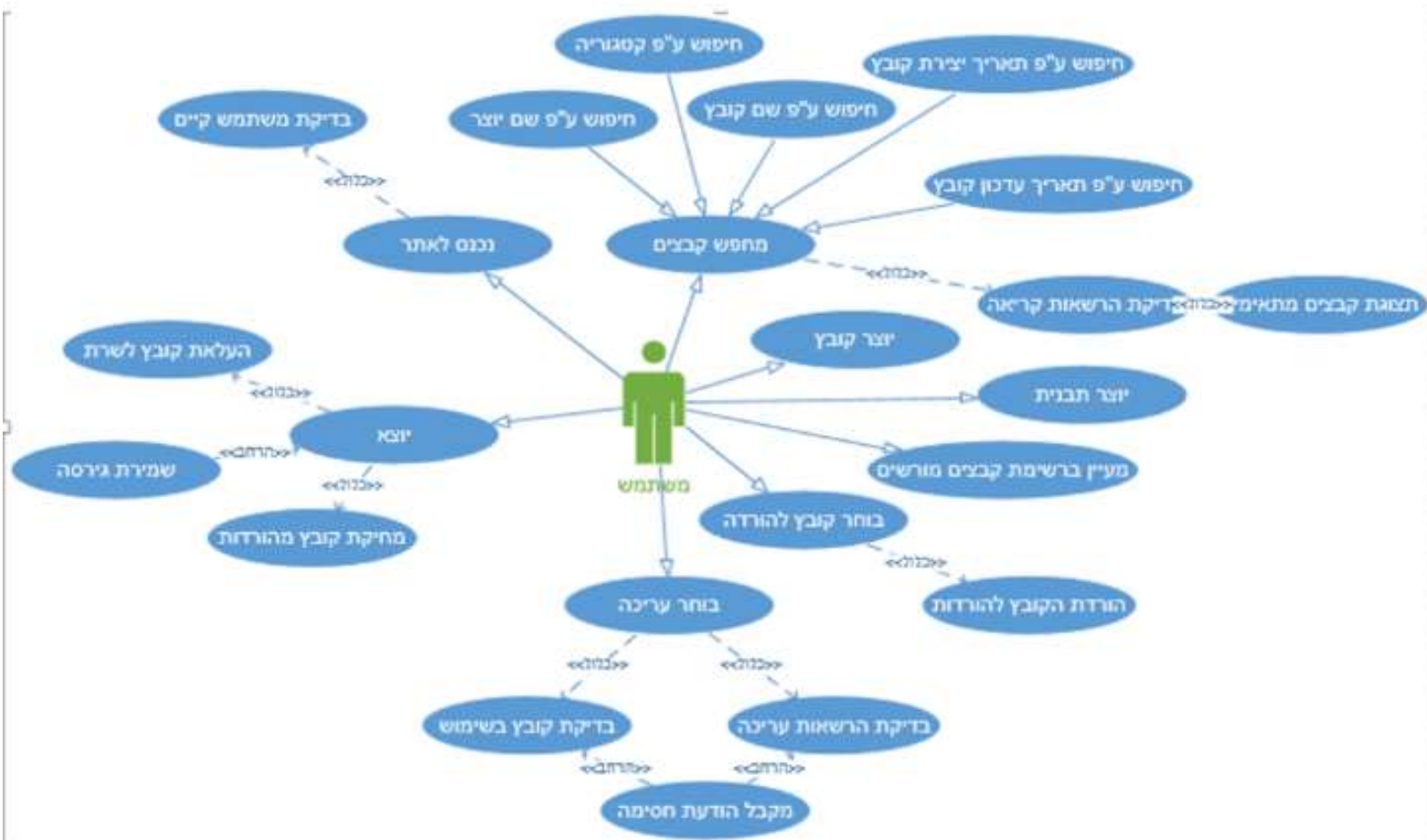
### **7.3 נא ציין מנגנוני התאוששות מנפילה/קריסה/תמיכה בטרנזקציות.**

Sql Server בנוי לתמיכה במס' גדול של קריאות בו זמניות ואין חשש לקריסה ברמה

של הפרויקט.

## 8. תרשימי מערכת מרכזיים

Use Case 8.1



8.2 Sequence diagram - רצף קריאות פונקציות מרכזיות בלוגיקה העיסקית המרכזית של הפרויקט

8.3 Data flow

## 9. תיאור המרכיב האלגוריתמי – חישובי:

9.1 איזה בעיה בא לפתור, איך יפתור?

9.2 איסוף מידע וניתוחים סטטיסטיים (אנליטיקות)

ניתן לבצע סטטיסטיקות על חיפוש הקבצים, השנים/החודשים שעליהם נעשים החיפושים הרבים ביותר.

מהם אחוזי השימוש בכל אחת מתוכנות אופיס. וכו',  
וכן ניתן לבצע סטטיסטיקה על אחוז המשתמשים שפעילים באתר בתדירות מסוימת.

### **10. תיאור/התייחסות לנושאי אבטחת מידע:**

נא לציין אזורים הדורשים אבטחה, כגון: שרת, בקרת גישה לאתר, חשבונות משתמשים, מאגרי מידע וכיצד ניתן מענה.

**גישה לאתר** - האתר יושב במרחב הרשת אך לא תינתן אפשרות גלישה אליו למי שאינו משתמש רשום.

**חשבונות משתמשים** - על מנת להיכנס לאתר יש צורך בסיסמא ושם משתמש.

**הרשאות גישה** - על מנת שהקבצים באתר יהיו מוגנים. ולא יתבצעו בהם שינויים או מחיקות, יינתנו הרשאות גישה למשתמשים בהתאם לצורך. כל משתמש ברמת הרשאה שלה הוא נצרך לצורך תפקידו באירגון, כגון: קריאה בלבד/עריכה/מחיקה וכו'.

### **11. נא ציין מס' מקרים ותגובות להם ניתן מענה אבטחתי.**

**מקרה:** המערכת מכירה בהיררכיה של עובדים, ואינה מעוניינת לאפשר עריכה/מחיקה של קבצים למשתמשים שלא קיבלו אישור לכך.

**תגובה:** מי שצרך את המשתמש לפרויקט נתן לו הרשאה "לקריאה בלבד" וכך מונע ממנו אפשרות לשנות את הקובץ.

**מקרה:** אדם פרטי שאינו מחברי הארגון עלול לגלוש לאתר ולצפות בקבצי הארגון.

**תגובה:** בעת כניסה לאתר יש להזין שם משתמש וסיסמא, וכך רק חברי הארגון יוכלו לצפות בקבצים.

### **12. משאבית הנדרשים לפרויקט:**

#### **12.1 מספר שעות המוקדש לפרויקט**

מספר השעות המוקדשות לפרויקט הוא 720.

#### **12.2 חלוקת עבודה בין חברי הצוות.**

360 שעות כל אחת.

#### **ציוד נדרש מחשב הכולל:**

CPU i5, RAM 8GB, HD SSD , חיבור לאינטרנט

#### **12.3 תוכנות נדרשות:**

Visual Studio , SQL Server, iis express , דפדפנים

#### **12.4 ידע חדש שנדרש ללמוד לצורך ביצוע הפרויקט:**

#### **12.5 ספרות ומקורות מידע:**

- Stackoverflow.com
- Codepan.com
- Msdn.microsoft.com
- Getbootstrap.com
- W3schools.com



### **13. תוכנית עבודה ושלבבים למימוש הפרויקט:**

#### **תוכנית עבודה ושלבבים למימוש הפרויקט:**

- 13.1 ייזום - יוני
- 13.2 ניתוח מערכת – מאי יוני
- 13.3 ניתוח מבנה נתונים – מאי יוני
- 13.4 אפיון UI – UX – יולי אוגוסט
- 13.5 כתיבת הלוגיקה העסקית – נובמבר דצמבר
- 13.6 כתיבת ממשק המשתמש – יולי אוגוסט
- 13.7 עיצוב - אוגוסט
- 13.8 בדיקות התוכנה - ינואר
- 13.9 התקנה והטמעה – ינואר פברואר

### **14. תכנון הבדיקות שיבוצעו:**

**14.1 נא פרט בטבלה, בדיקות תהליכיות ברמת משתמש בהן נדרשת המערכת**

**לעמוד (full flow)**

**14.2 נא פרט בטבלה, מס מייצג של בדיקות יחידה למודולים המרכזיים בהן**

**נדרשת המערכת לעמוד (unit test)**

מס' בדיקה	מס' דרישה במסמך איפיון	מקרי הבדיקה	ידנית/אוטומטית	חשיבות	הערות
1	4.2.1	יש להקיש שם משתמש וסיסמה ולראות שקיים ואם לא המערכת שולחת הודעה ע"כ.	ידנית	גבוהה	
2	4.2.1	יש לבדוק שאורך הסיסמה תקין	ידנית	גבוהה	
3	4.2.2	לבדוק את כל אפשרויות החיפוש ושמתקבלות אכן תוצאות נכונות.	ידנית	גבוהה	
4	4.2.2	המערכת מציגה קבצים ע"פ הסינון של המשתמש.	ידנית	גבוהה	
5	4.2.3	לבדוק שבשעת לחיצה על יצירת קובץ ובחירת סוג הקובץ נוצר קובץ מתאים בתקיית ההורדות של המשתמש.	ידנית	גבוהה	
6	4.2.4	לבדוק שבשעת לחיצה על יצירת תבנית ובחירת סוג הקובץ נוצר קובץ מתאים בתקיית ההורדות של המשתמש <sup>9</sup>	ידנית	גבוהה	

	גבוהה	ידנית	הקובץ הנוכחי יורד לתקיית Download/ההורדות.	4.2.7	<b>7</b>
	גבוהה	ידנית	לבדוק כאשר הקובץ פתוח במחשב אחר שהקובץ מורשה בקריאה בלבד.	4.2.8	<b>8</b>
	גבוהה	ידנית	לבדוק שהקובץ מורשה בעריכה ע"פ ההרשאות, ואם לא המערכת מודיעה ע"כ למשתמש.	4.2.9	<b>9</b>
	בינונית	ידנית	לבדוק מקרה של שמירה בגירסה מתקדמת במקרה שהמשתמש מעוניין ויש לו הרשאת עריכה.	4.2.10	<b>10</b>
	גבוהה	ידנית	לבדוק מחיקה של הקובץ מתקיית Download/ההורדות.	4.2.10	<b>11</b>
	בינונית	ידנית	לבדוק שמנהל המערכת מקבל הרשאות מלאות.	4.2.11	<b>12</b>

חתימת המנחה האישי

חתימת הסטודנט

ג. הערות ראש המגמה במכללה

---

---

---

ד. אישור ראש המגמה

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

ה. הערות הגורם המקצועי מטעם מה"ט

---

---

---

ו. אישור הגורם המקצועי מטעם מה"ט

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

## קווים מנחים בבחירת נושאים והיקפי עבודה בפרויקט הגמר.

1. דגשים ארכיטקטוניים ושיקולים במימוש:
  - 1.1. מומלץ להתנסות בארכיטקטורות השלבות שימוש בתצורת שרת לקוח.
  - 1.2. שימוש ב- design patterns במודולי התוכנה השונים- באיזורים מתאימים.
  - 1.3. דגש על הפרדה בין לוגיקה עסקית השייכת לצד לקוח וצד שרת. (FrontEnd, Backend, ServerBL, ClientBL)
  - 1.4. חלופות ארכיטקטוניות נדרשות לתמוך או לספק מענה לדרישות כגון:
    - 1.4.1. תמיכה והתמודדות בוויסות עומסים .
    - 1.4.2. תמיכה והתמודדות עם שיקולי אבטחה והגנה על מידע.
    - 1.4.3. תמיכה בשרידות והתאוששות מתקלה(טרנזקציות שמירה למאגר, והתאוששות)
  - 1.5. תמיכה בשיקולי חווית משתמש (צד מנהל מערכת וצד משתמש קצה)
  - 1.6. תמיכה היכן שניתן בניהול פרופילי משתמשים.
2. שפת מימוש הפרויקט-
 

ישנו משקל גבוה במימוש הפרויקט ביותר משפת מימוש אחת לפרויקט, תוך מתן דגש ליתרונות היחסיים של כל שפה, עבור מודול תוכנה במכלולי הפרויקט. למשל במקרה של תצורת שרת לקוח (אתר אינטרנט):

  - 2.1.1. לצד הניהול העיסקי של השרת, בחירה בשפות עיליות JAVA, C# או nodeJS.
  - 2.1.2. לתכולה חישובית/אלגוריתמית- מימוש בשפת native נניה C++, C.
  - 2.1.3. לצד לקוח AngularJS, Asp.net וכך
3. מאגר נתונים Database:
  - 3.1. ישנה חשיבות גבוהה להתנסות בעבודה עם מאגרי נתונים למשל, מאגר רלציוני ומאגר FS Based למשל:
    - 3.1.1. עבור מאגר רלציוני נבחר ב- 'Sql server, Sqlite, etc'
    - 3.1.2. עבור מאגר לא רלציוני נבחר ב- mongoDB או NoSql .
  - 3.2. ישנה חשיבות רבה להגדרת שכבת גישה למאגר הנתונים כזו שתנהל מרכיבים טרנזקטיביים וסנכרון. נין להשתמש גם במסגרת frameworks קיימים כדוגמת dotNet.
4. מרכיב אלגוריתמי/ חישובי-
 

ישנה חשיבות רבה להתנסותו של התלמיד והתמודדותו עם יכולות חישוביות במסגרת מכלולי הפרויקט. ניתן לשלב היבטים אלגוריתמיים או לחילופין ניתוחים וחיתוכים סטטיסטיים בסיסיים מעל מאגר נתונים, למשל:

  - 4.1. במקרה של אתרים כניסת משתמשים, גיאומטריה חישובית וכך.

5. בדיקות תוכנה:
- 5.1. יש לגזור מדרישות המוצר אוסף בדיקות שיפורדו לפחות לשתי קטגוריות מרכזיות ויכסו את מרבית הקוד:
- 5.1.1. בדיקות יחידה (Unit-Test) - אלה הן אותן בדיקות אותן יממש המפתח ברמת פנים מכלולי התוכנה ועד לרמת הפונקציות הציבוריות באותם מכלולי תוכנה.
- 5.1.2. בדיקות תהליכיות (Full Flow) - הללו יעסקו בעיקר בבדיקות בקשר שבין מכלולי תוכנה מרכזיים ויבחנו את הפונקציונאליות האינטגרטיבית של המוצר, מקצה לקצה.
- 5.1.3. גישות מבורכות לתהליך ניתן לאמץ מתוך גישות שונות
- 5.1.4. למשל:
- 5.1.4.1. code a little test a little
- 5.1.4.2. test driven development, ,
- 5.1.4.3. Regression Tests

### פרקים מרכזיים נוספים לספר הפרויקט:

13. דרישות מערכת ופונקציונאליות
- 13.1. הנחות יסוד
- 13.1.1. הנחות יסוד הקשורות לסביבה הטכנולוגית ולתפקוד .
- 13.2. דרישות מערכת
- סביבת הטמעה ושימוש. שרידות, ביצועים/התמודדות עם עומסים.
- 13.3. דרישות פונקציונאליות.
- רשימת דרישות המשתמש מהמערכת, מהן הפעולות בהן נדרשת המערכת לתמוך.
- 5.2. חלופות ארכיטקטוניות-
- דיון בבחירת ארכיטקטורות ברמת המערכת וברמת מכלולי התוכנה.
- 5.2.1. ברמת המערכת- חשוב להציג בספר הפרויקט בחינה של מספר חלופות ארכיטקטוניות (לפחות 3) בהן יבחן התלמיד את האפשרויות השונות תחת מספר אילוצים רלוונטיים נשוא הפרויקט, בין יתר השיקולים ניתן לשקול: התמודדות עם עומסים וויסותם, שרידות, יכולת מימוש, זמינות טכנולוגית ועוד. יש לשקול שילוב במקומות המתאימים של design patterns מקובלים, הן ברמת המערכת והן ברמת מכלולי התוכנה.
- 5.2.2. ברמת מכלולי התוכנה- ניתן להציג דיון חלופות מצומצם יותר, אך לשקול בחיוב שילוב של design patterns מקובלים במקומות המתאימים (להימנע משימוש מיותר).
6. טופולוגיית הפתרון הנבחר- הצגה סכימתית של פרישת המערכת.
7. ארכיטקטורה נבחרת: הצגה בגישת UML את פריסת מרכיבי הפתרון בחלוקה למכלולי תוכנה ראשיים ומשניים כמו-גם הדיאגרמות הרלוונטיות.
- 7.1. שימוש במבני נתונים וארגון קבצים
- 7.1.1. נא פרט את מבני הנתונים.
- 7.1.2. נא פרט את שיטת האיחסון (מאגר, קבצים ובאיזה טכנולוגיה)
- 7.1.3. נא ציין מנגנוני התאוששות מנפילה/קריסה/התמיכה בטראנזקציות.

7.2. תרשים זרימת המידע במערכת

Use Cases 7.2.1.

Sequence diagram 7.2.2

Data flow 7.2.3

7.3. חלופות שפת מימוש-

במסגרת ספר הפרויקט חשוב להציג בחינה של מספר חלופות עבור שפה/ות מימוש הפרויקט. הנ"ל צריך לכלול דרישות אותן יגדיר התלמיד בבחירת השפה המתאימה. בין יתר השיקולים ניתן לכלול: זמני ריצה, היבטי אבטחה והגנה, הגנה על זכויות יוצרים (בינארי או interpreter, קלות במימוש, התאמה לממשקי משתמש או צד שרת) וכף.

8. חלופות אבטחתיות והגנה -

במסגרת ספר הפרויקט יש לבחון לפחות 3 חלופות אבטחתיות להגנה ושמירה על נתונים, יש לכלול התמודדות עם מקרים ותגובות בתהליך הבחינה (לפחות 10 מקרים).

8.1.1 יש לשים לב- לפרוטוקולי תקשורת, http, https, ssl###,

8.1.2 מכלולי תוכנה צד שלישי (אנטי וירוס, מצפינים, firewalls וכף).

8.1.3 ברמת הקוד- ווידוא סכימות הודעות בין מכלולי התוכנה ובתקשורת בניהם.

8.2. פירוט בדיקות תוכנה ואופן ביצוען (STP-DOC) - לכלול את רשימת בדיקות התוכנה, בדיקות יחידה, בדיקות תהליכיות- full Flow במסגרת מסמך תכנון בדיקות ובדיקות. הנ"ל יוצג בטבלה : תיאור הבדיקה, תוצאה רצויה, תוצאה מתקבלת.

ניתן לאמץ את נוהל מפתח באופן מושפל!

[http://www.methodacloud.com/content/pages/kit\\_maxsum/H\\_Guide-map.asp](http://www.methodacloud.com/content/pages/kit_maxsum/H_Guide-map.asp)