

תאריך: _____

לכבוד
יחידת הפרויקטים
מה"ט

הצעה לפרויקט גמר

* יש להדפיס את כל הנתונים הנדרשים

א. פרטי הסטודנטים

שם הסטודנט	ת.ז. 9 ספרות	כתובת	טלפון נייד	תאריך סיום הלימודים

שם המכללה _____ סמל המכללה: _____

מסלול ההכשרה: הנדסאים/טכנאים מוסמכים (מחק את המיותר).

מגמת לימוד: _____ מקום ביצוע הפרויקט: _____

ב. פרטי המנחה האישי

שם המנחה *	כתובת	טלפון נייד	תואר	מקום עבודה/תפקיד

* עבור מנחה אישי חדש יש לצרף קורות חיים, ניסיון מקצועי ותעודות השכלה לאישור מה"ט.

חתימת הסטודנט _____ חתימת המנחה האישי _____ חתימת הגורם המקצועי מטעם מה"ט _____

[Type here]

1. שם הפרויקט

2. רקע

2.1. תיאור ורקע כללי

2.2. מטרות המערכת

3. סקירת מצב קיים בשוק, אילו בעיות קימות

4. מה הפרויקט אמור לחדש או לשפר

5. דרישות מערכת ופונקציונאליות

5.1. דרישות מערכת

סביבת הטמעה ושימוש. שרידות, ביצועים/התמודדות עם עומסים.

5.2. דרישות פונקציונאליות

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

6. בעיות צפויות במהלך הפיתוח ופתרונות (תפעוליות, טכנולוגיות, עומס ועוד):

6.1. תיאור הבעיות- הללו כפועל יוצא של דרישות המשתמש מהתוכנה.

6.2. פתרונות אפשריים. (נא ציין פתרונות אפשריים וחלופות ארכיטקטוניות)

7. פתרון טכנולוגי נבחר :

7.1. טופולוגיית 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 וכו')

7.2. טכנולוגיות בשימוש. (איזה ומדוע בכמה מילים)

7.3. שפות הפיתוח : (איזה שפות ומדוע בכמה מילים?)

7.4. תיאור הארכיטקטורה הנבחרת- הסבר בכמה מילים מדוע

7.5. חלוקה לתכניות ומודולים

7.6. סביבת השרת (מקומי, וירטואלי, ענן, שירות אירוח)

7.7. ממשק המשתמש/לקוח - GUI

7.8. ממשקים למערכות אחרות / API :

7.9. שימוש בחבילות תוכנה :

8. שימוש במבני נתונים וארגון קבצים

8.1. נא פרט את מבני הנתונים.

8.2. נא פרט את שיטת האיחסון (מאגר, קבצים ובאיזה טכנולוגיה)

8.3. נא ציין מנגנוני התאוששות מנפילה/קריסה/התמיכה בטראנזקציות.

9. תרשימי מערכת מרכזיים

9.1. Use Case

9.2. Sequence diagram - רצף קריאות פונקציות מרכזיות בלוגיקה העיסקית המרכזית של

הפרוייקט

9.3. Data flow

10. תיאור המרכיב האלגוריתמי – חישובי

10.1. איזה בעיה בא לפתור, איך יפתור?

10.2. איסוף מידע וניתוחים סטטיסטיים (אנליטיקות)

11. תיאור/התייחסות לנושאי אבטחת מידע

נא לציין אזורים הדורשים אבטחה, כגון: שרת, בקרת גישה לאתר, חשבונות משתמשים, מאגרי מידע וכיצד ניתן מענה.
נא ציין מס' מקרים ותגובות להם ניתן מענה אבטחתי.

12. משאבים הנדרשים לפרויקט:

12.1. מספר שעות המוקדש לפרויקט, חלוקת עבודה בין חברי הצוות

12.2. ציוד נדרש

12.3. תוכנות נדרשות

12.4. ידע חדש שנדרש ללמוד לצורך ביצוע הפרויקט

12.5. ספרות ומקורות מידע

13. תכנית עבודה ושלבנים למימוש הפרויקט

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)

15. בקרת גרסאות (version control)

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: _____

נספח להצעת הפרויקט:

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 נניח ++C, C.
 - 2.1.3. לצד לקוח Asp.net, AngularJS וכו'

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 וכו'.

5. בדיקות תוכנה:

5.1. יש לגזור מדרישות המוצר אוסף בדיקות שיופרדו 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:

16. דרישות מערכת ופונקציונאליות

16.1. הנחות יסוד

16.1.1. הנחות יסוד הקשורות לסביבה הטכנולוגית ולתפקוד.

16.2. דרישות מערכת

סביבת הטמעה ושימוש. שרידות, ביצועים והתמודדות עם עומסים.

16.3. 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 הקריסה והתמיכה בטראנזקציות.

7.2. תרשים זרימת המידע במערכת

7.2.1. Use Cases

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 שפהות 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) וכו'.

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 וכו').

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 מושכל!

http://www.methodacloud.com/content/pages/kit_maxsum/H_Guide-map.asp