תאריך: \_\_\_\_\_\_\_\_\_\_\_

### לכבוד

### יחידת הפרויקטים

מה"ט

### **הצעה לפרויקט גמר**

\* יש להדפיס את כל הנתונים הנדרשים

1. **פרטי הסטודנטים**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| שם הסטודנט | ת.ז. 9 ספרות | כתובת | טלפון נייד | תאריך סיום הלימודים |
| רגד אבו רחאל | 212809149 | ריינה | 0525385082 | 10/24 |
|  |  |  |  |  |

שם המכללה \_ מכללה טכנולוגית כנרת \_\_\_\_\_\_\_\_\_\_\_ סמל המכללה: \_\_\_72228\_\_\_\_\_

מסלול ההכשרה: הנדסאים.

מגמת לימוד: \_תוכנה\_\_\_\_\_\_\_\_\_ מקום ביצוע הפרויקט: \_\_ מכללה טכנולוגית כנרת

**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

# שם הפרויקט

מערכת ניהול מלאי

# רקע

# תיאור ורקע כללי

מערכת לניהול מלאי מספקת מידע מפורט ומעודכן על המלאי באופן אקטיבי

# מטרות המערכת

המטרה העיקרית של המערכת היא לשפר את ניהול המלאי על ידי יכולת צפייה וניתוח מדויקים של המלאי הקיים, אוטומציה של תהליכי ההזמנה והמשלוח, פעולה מראש לפי צרכי הלקוח, וקידום יעילות בניהול המלאי כולל חיסכון בעלויות והגברת התפוקה.

# סקירת מצב קיים בשוק, אילו בעיות קימות

בשוק הנוכחי, מרבית מערכות ניהול המלאי חסרות יכולת עדכון אוטומטי, חסרות יכולת אנליטית מתקדמת ואינן מספקות ממשק ידידותי ומותאם אישית למשתמש. בעיות נוספות כוללות חוסר בדיוק בנתוני המלאי וקשיים בניהול הזמנות ומלאי באופן אפקטיבי.

# מה הפרויקט אמור לחדש או לשפר

# הפרויקט מיועד לפתרון הבעיות המזוהות בניהול המלאי הנוכחי. זה יכלול יישום מערכת חדשה ומתקדמת שתאפשר ניהול מלאי אופטימלי ויעיל יותר, עם דגש על עדכון רציף של מידע, אוטומציה של תהליכים ואנליזת נתונים מתקדמת לקבלת החלטות מבוססות נתונים.

# דרישות מערכת ופונקציונאליות

# דרישות מערכת סביבת הטמעה ושימוש. שרידות, ביצועים\התמודדות עם עומסים. סביבת ההטמעה שרת וחיבור לרשת האינטרנט מערכת עם גישה לפי משתמש והרשאות כניסה תלוי בסוג הפעילות

# דרישות פונקציונאליות רשימת דרישות המשתמש מהמערכת, מהן הפעולות בהן נדרשת המערכת לתמוך. המשתמש מקבל שם משתמש וסיסמה למערכת לפי דרישה לרישיונות למשתמשים המשתמש מוגבל בהרשאות פעילות וכך מנהל המערכת תהיה לא כול ההרשאות וכך גם מתאפשר לשמור על יותר מידע מחשיפה לעובדים שאין להם את האישור לגישה מידע הזה

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

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

-אי תאימות בין התוכנה הפועלת לדרישות המשתמש או לתקנות תעשייתיות.

-בעיות בביצועים וזמני תגובה עקב התמודדות עם עומסים גדולים.

-בעיות אבטחה כתוצאה מתקפות רשת או פרצות במערכת.

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

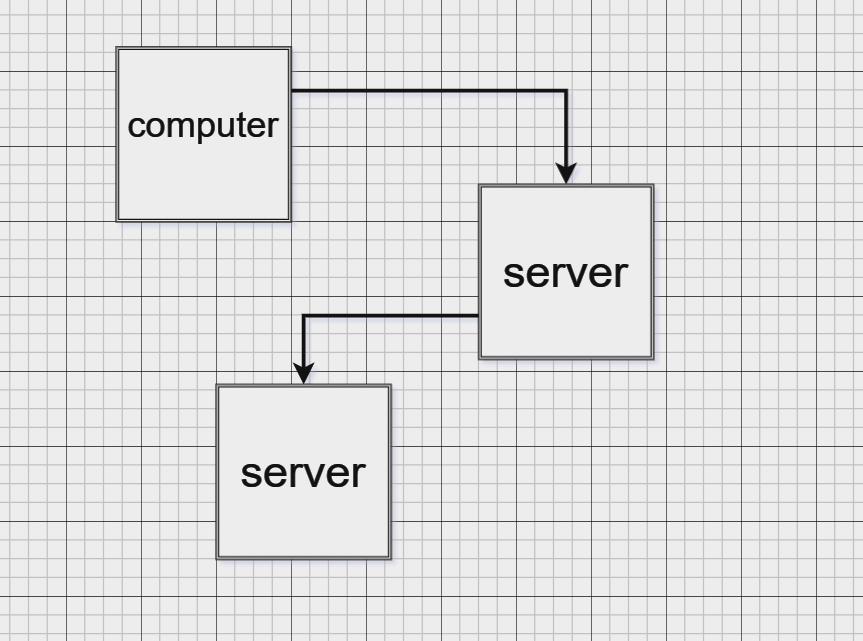
-בכדי לפתור אי התאמות בין התוכנה לדרישות המשתמש, יש להבטיח כי תהליכי הפיתוח והבדיקה מתבצעים בהתאם לדרישות המשתמש ולתקנים תעשייתיים.

-לטפח תפיסה של "סקרנות ובדיקה" בתהליכי הפיתוח על מנת לזהות בעיות בביצועים ולפתור אותן במהירות.

-לממש מנגנוני אבטחה חזקים כגון הגנה בפני תקפים XSS ו-CSRF וכן לעדכן באופן תדיר את התקנות האבטחה ולבצע בדיקות פנימיות וחיצוניות כדי לזהות ולטפל בפרצות פוטנציאליות.

# פתרון טכנולוגי נבחר:

# 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 תקשורתי-אינטרנט, המערכת תיושם ברשת האינטרנט , יש להציג את דיאגרמת המערכת וכו')



# טכנולוגיות בשימוש.(איזה ומדוע בכמה מילים)

טכנולוגיות בשימוש WINFORM בשפת C# ובסיסי נתונים SQL SERVER

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

C# היא שפת תכנות שפותחה על ידי מיקרוסופט, והיא נמצאת בשימוש נרחב בעיקר לפיתוח יישומים בסביבת ה.NET. השפה מתמקדת בבטיחות, ביציבות ובביצועים גבוהים. C# נוצרה כחלק מפלטפורמת הפיתוח .NET ומספקת יכולות רבות לפיתוח יישומים שונים כמו אפליקציות משתמש, אפליקציות עסקיות, ושירותי רשת.

היתרון המרכזי של C# הוא השמתקדותה בכלים ובתשתיות המסופקות על ידי מיקרוסופט, כך שמשתמשי C# יכולים להפיק מועדונים רבים מהיתרונות של סביבת הפיתוח של .NET. השפה תומכת בתכנות מונחה עצמים (OOP) ומספקת תמיכה מובנית לפיתוח יישומים רבי-לתחומים ומורכבים.

# תיאור הארכיטקטורה הנבחרת- הסבר בכמה מילים מדוע

תוכנה שתאפשר לשתף מידע בין גופים שונים

# חלוקה לתכניות ומודולים

המערכת מחולקת למודולים ראשיים הכוללים ניהול מלאי, ניהול הזמנות, ניהול לקוחות, ניהול ספקים, ודוחות וניתוחים. כל מודול משתמש במתודולוגיה של מחלקות ופונקציות נפרדות על מנת להפחית קוד כפול ולהפוך את הקוד לנקי וניתן לתחזוקה.

# סביבת השרת (מקומי, וירטואלי, ענן, שירות אירוח)

המערכת תיושם בסביבה של שרת מקומי עם חיבור לרשת האינטרנט, ויתכן כי יש להשתמש בשירותי אירוח לשמירה ותחזוקת המידע.

# ממשק המשתמש/לקוח – GUI

# ממשקים למערכות אחרות /:API

אין ממשקים מפותחים למערכות אחרות בשלב זה, אך ישנה אפשרות לפיתוח API לשילוב עתידי עם מערכות חיצוניות בהמשך.

# שימוש בחבילות תוכנה: המערכת משתמשת בחבילות תוכנה נוספות כגון Entity Framework לניהול וגישה למסדי הנתונים, ו-NUnit לבדיקות יחידה.

# שימוש במבני נתונים וארגון קבצים

# נא פרט את מבני הנתונים.

A screenshot of a computer

Description automatically generated

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

מאגר בסיסי נתונים mysql

# נא ציין מנגנוני התאוששות מנפילה\קריסה\תמיכה בטראנזקציות.

רישום (Logging): המערכת משתמשת ברישום לשמירת פעולות המבוצעות ולאפשר תיעוד ובדיקה של המצב של המערכת לאחר חזרה מנפילה או קריסה.

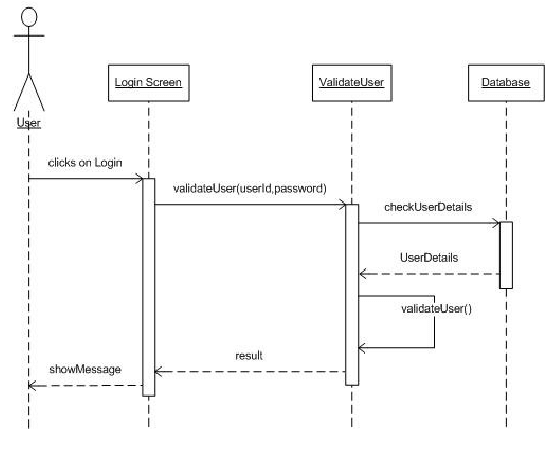
טרנזקציות (Transactions): המערכת משתמשת בטרנזקציות כדי להבטיח קבלת מצב תקין לאחר החזרה מנפילה או קריסה. זה כולל נעילת משאבים וביצוע פעולות בצורה אטומית.

גיבוי (Backup): המערכת משתמשת בגיבויים כדי להבטיח שנתוני המערכת יוכלו להשתקם לאחר נפילה או קריסה. הגיבויים מאוחסנים במקום מתאים ובתדירות שתוכננה מראש

# תרשימי מערכת מרכזיים

# A diagram of a flowchart Description automatically generatedUse Case אופן השימוש

# Sequence diagram - רצף קריאות פונקציות מרכזיות בלוגיקה העיסקית המרכזית של הפרוייקט



# Data flow

במערכת הניהול של מלאי, הנתונים מתחילים בזרימה ממקורות שונים במערכת. הזרימה מתחילה עם רישום פריטי המלאי, כמו מוצרים, כמויות זמינות, וכמויות נכונות למשלוח. נתונים אלו מתקבלים מספקים, מפעילים או ממערכות חיצוניות.

הנתונים עוברים לשלב העיבוד, בו מתבצעת עיבוד וניתוח של הנתונים. זה כולל עדכון נתונים על בסיס יומי, ניתוח טרנדים במלאי, וזיהוי צרכים ופטורים במלאי.

לאחר מכן, הנתונים עוברים לשלב האחסון והניהול, שבו הם מאוחסנים במאגר הנתונים המרכזי של המערכת. כאן, הנתונים מסודרים, מטופלים, ומסונכרנים על מנת לשמור על תמיכה ונגישות אופטימלית.

לבסוף, הנתונים נגישים למשתמשים דרך ממשק המשתמש, כאשר הם יכולים לצפות במלאי זמין, לבצע הזמנות חדשות, ולבצע פעולות ניהוליות נוספות בהתאם להרשאותיהם.

# תיאור המרכיב האלגוריתמי – חישובי

# איזה בעיה בא לפתור, איך יפתור?

הבעיה המרכזית היא ניהול המלאי באופן יעיל ואופטימלי. זה כולל ניהול תהליכי הזמנה וספק, ניהול מלאי קיים וכן אנליזת נתונים לצורך קבלת החלטות מבוססות נתונים. פתרון עוסק בפיתוח מערכת מתקדמת המספקת אוטומציה ואנליזת נתונים מתקדמת לניהול מלאי.

# איסוף מידע וניתוחים סטטיסטיים (אנליטיקות)

המערכת מאפשרת איסוף נתונים מרובים ומקורות שונים במערכת, ובאמצעות כלים אנליטיים מתקדמים ניתן לבצע ניתוחים סטטיסטיים על מנת להבין את המגמות והתהליכים העיקריים בניהול המלאי ובתהליכי הזמנה וספק.

# 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.

נקודות האבטחה המרכזיות כוללות שמירה על נתוני הלקוחות, המלאי והתהליכים באמצעות ניהול גישה מאובטח, הצפנת נתונים במסדי הנתונים, בקרת גישה מבוקרת וניטור פעילות תוך שמירה על תקינות המערכת. מתוך כך, ניתן להשתמש במנגנונים כגון SSL/TLS להצפנת התקשורת, גישה מבוקרת למערכת, והגבלת גישה לנתונים בהתאם לרמות ההרשאה.

# משאבים הנדרשים לפרויקט: ????????

# מספר שעות המוקדש לפרויקט, חלוקת עבודה בין חברי הצוות

# ציוד נדרש

# תוכנות נדרשות

# ידע חדש שנדרש ללמוד לצורך ביצוע הפרויקט

# ספרות ומקורות מידע

# 

# תכנית עבודה ושלבים למימוש הפרויקט

# נקודות האבטחה המרכזיות כוללות שמירה על נתוני הלקוחות, המלאי והתהליכים באמצעות ניהול גישה מאובטח, הצפנת נתונים במסדי הנתונים, בקרת גישה מבוקרת וניטור פעילות תוך שמירה על תקינות המערכת. מתוך כך, ניתן להשתמש במנגנונים כגון SSL/TLS להצפנת התקשורת, גישה מבוקרת למערכת, והגבלת גישה לנתונים בהתאם לרמות ההרשאה.

# תכנון הבדיקות שיבוצעו

# נא פרט בטבלה, בדיקות תהליכיות ברמת משתמש בהן נדרשת המערכת לעמוד (full Flow).

|  |  |  |  |
| --- | --- | --- | --- |
| מס בדיקה | שם בדיקה | מה עשיתי הבדיקה | האם עבר או לא |
| 1 | כניסה למערכת | בדיקה אם הדף כניסה ניגש לבסיסי נתונים ונכנס לעמוד ראשי | כן |
| 2 | כניסה למערכת האחסנה | בדיקה אם הדף כניסה ניגש לבסיסי נתונים ונכנס לעמוד ראשי | כן |

# נא פרט בטבלה, מס מייצג של בדיקות יחידה למודולים המרכזיים בהן נדרשת המערכת לעמוד. (unit test)

|  |  |  |  |
| --- | --- | --- | --- |
| בדיקת קבלת משתמש חדש | מצד לקוח : כן | צד שרת : כן | בסיסי נתונים : כן |
| בדיקת שאילתות SQL | כן מציג ב SELECT | כן מכניס ב INSERT | מציג נתונים ב צד לקוח |
| בדיקת כניסה למערכת | קבלת שם וסיסמה בצד לקוח | צד שרת: מאמת נתונים עם בסיסי נתונים | אחרי אימות אם הוא עובר לדף ראשי שהנתונים נכונים ואם מציג הודעה אם שגוי |

# בקרת גרסאות (version control)

עבודה עם 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: \_\_\_\_\_\_\_\_\_\_\_\_\_

###### 

**נספח להצעת הפרויקט:**

**קווים מנחים בבחירת נושאים והיקפי עבודה בפרויקט הגמר.**

1. **דגשים ארכיטקטוניים ושיקולים במימוש:**
   1. מומלץ להתנסות בארכיטקטורות השלבות שימוש בתצורת שרת לקוח.
   2. שימוש ב- design patterns במודולי התוכנה השונים- באיזורים מתאימים.
   3. דגש על הפרדה בין לוגיקה עסקית השייכת לצד לקוח וצד שרת.(FrontEnd,Backend ,ServerBL ,ClientBL)
   4. חלופות ארכיטקטוניות נדרשות לתמוך או לספק מענה לדרישות כגון:
      1. תמיכה והתמודדות בוויסות עומסים .
      2. תמיכה והתמודדות עם שיקולי אבטחה והגנה על מידע.
      3. תמיכה בשרידות והתאוששות מתקלה(טרנזקציות שמירה למאגר, והתאוששות)
   5. תמיכה בשיקולי חווית משתמש (צד מנהל מערכת וצד משתמש קצה)
   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):
   * 1. לצד הניהול העיסקי של השרת, בחירה בשפות עיליות JAVA, C# או nodeJS.
     2. לתכולה חישובית\ אלגוריתמית- מימוש בשפת native נניח C, C++.
     3. לצד לקוח AngularJS,Asp.net וכו'
3. **מאגר נתונים Database:**
   1. ישנה חשיבות גבוהה להתנסות בעבודה עם מאגרי נתונים למשל, מאגר רלציוני ומאגר FS Based.   
      למשל:
      1. עבור מאגר רלציוני נבחר ב- Sql server, Sqlite, etc’
      2. עבור מאגר לא רלציוני נבחר ב- mongoDB או NoSql.
   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:
   1. במקרה של אתרים כניסת משתמשים, גיאומטריה חישובית וכו'.
5. **בדיקות תוכנה:**
   1. יש לגזור מדרישות המוצר אוסף בדיקות שיופרדו לפחות לשתי קטגוריות מרכזיות ויכסו את מרבית הקוד:
      1. בדיקות יחידה (Unit-Test)- אלה הן אותן בדיקות אותן יממש המפתח ברמת פנים מכלולי התוכנה ועד לרמת הפונקציות הציבוריות באותם מכלולי תוכנה.
      2. בדיקות תהליכיותFull Flow) – הללו יעסקו בעיקר בבדיקות בקשר שבין מכלולי תוכנה מרכזיים ויבחנו את הפונקציונאליות האינטגרטיבית של המוצר, מקצה לקצה.
      3. גישות מבורכות לתהליך ניתן לאמץ מתוך גישות שונות
      4. למשל:
         1. code a little test a little
         2. , test driven development,
         3. Regression Tests

**פרקים מרכזיים נוספים לספר הפרויקט:**

# דרישות מערכת ופונקציונאליות

# הנחות יסוד

# הנחות יסוד הקשורות לסביבה הטכנולוגית ולתפקוד .

# דרישות מערכת סביבת הטמעה ושימוש. שרידות, ביצועים\התמודדות עם עומסים.

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

* 1. **חלופות ארכיטקטוניות-**   
     דיון בבחירת ארכיטקטורות ברמת המערכת וברמת מכלולי התוכנה.
     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.
     2. ברמת מכלולי התוכנה- ניתן להציג דיון חלופות מצומצם יותר, אך לשקול בחיוב שילוב של design patterns מקובלים במקומות המתאימים (להימנע משימוש מיותר).

1. **טופולגיית הפתרון הנבחר**- הצגה סכימתית של פרישת המערכת.

# ארכיטקטורה נבחרת: הצגה בגישת UML את פריטת מרכיבי הפתרון בחלוקה למכלולי תוכנה ראשיים ומשניים כמו-גם הדיאגרמות הרלוונטיות.

# שימוש במבני נתונים וארגון קבצים

# נא פרט את מבני הנתונים.

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

# נא ציין מנגנוני התאוששות מנפילה\קריסה\תמיכה בטראנזקציות.

# תרשים זרימת המידע במערכת

# Use Cases

# Sequence diagram

# Data flow

* 1. **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) וכו'.

1. **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).
   * 1. יש לשים לב- לפרוטוקולי תקשורת, http ,https ,ssl###,
     2. מכלולי תוכנה צד שלישי (אנטי וירוס,מצפינים,firewalls וכו').
     3. ברמת הקוד- ווידוא סכימות הודעות בין מכלולי התוכנה ובתקשורת בניהם.
   1. **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**](http://www.methodacloud.com/content/pages/kit_maxsum/H_Guide-map.asp)