

תאריך: 03 / 12 / 24

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
דור אלחיאני	203933866	בית שאן, חיים וייצמן 17 ו'	050 – 9116602	2025

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

1. שם הפרויקט
HydroBuddy

2. רקע
2.1. תיאור ורקע כללי

ערכת השקיית צמחים הנשלטת ע"י לוח Arduino עם יכולת ניהול השקיית הצמח דרך אפליקציה .
הבעיה – אי נגישות / הימצאות בשטח בזמן שהצמח צריך לקבל השקיה

2.2. מטרות המערכת

ליצור פלטפורמה אחת הכוללת צד שרת וצד קידמי בעלת טכנולוגיה קיימת כדי לנגיש יותר אנשים לגידול צמחים בבתיים

[Type here]

3. סקירת מצב קיים בשוק, אילו בעיות קימות

ישנן פלטפורמות רבות להרכבה עצמית, אך השליטה נעשית דרך אפליקציה חיצונית שלא מתממשקת היטב עם כלל הרכיבים. יכולת הניטור, דליית המידע ואפיון כל צמח לוקות בחסר

4. מה הפרויקט אמור לחדש או לשפר

פלטפורמה אחידה מקצה אל קצה הנותנת למשתמש חוויה נעימה ואינדיקציה לגבי טיב הצמח ויכולת להתממשק עימו בצורה ידידותית

5. דרישות מערכת ופונקציונאליות

5.1. דרישות מערכת

סביבת הטמעה ושימוש. שרירות, ביצועים והתמודדות עם עומסים.

דרישות המערכת ליישום ושימוש כוללת חומרה מבוססת Arduino המשמש כשרת מסד נתונים ושליחת מידע. על גביה יהיו חיישנים המנתרים את טיב איכות הצמח. תצורת הפלטפורמה תתבצע על ידי שרת מקומי שמבטיח תקשורת אלחוטית. השרת הנמצא בענן מנהל עיבוד נתונים ויתחבר אל מסד הנתונים. תכונות השרידות כוללת טיפול בחוסר תקשורת במכשיר ה Arduino ואחסון נתונים כדי לטפל בשיבושי חומרה / רשת

5.2. דרישות פונקציונאליות

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

פונקציונליות בסיסיות :

למערכת תהיה שלושה מצבים (השקיה אוטומטית, השקיה ידנית, מצב שבת)

1. **מצב אוטומטי** – מצב בו ההשקיה מתבצעת באופן אוטומטי לפי תנאי מזג האוויר, לחות האדמה, אור השמש.
2. **מצב ידני** – מצב בו המשתמש שולט באופן ישיר על המשאבה וההשקיה של הצמח
3. **מצב שבת** – מצב דומה למצב האוטומטי, ההבדל העיקרי הוא שהמשתמש יוכל לקבוע את זמני ההשקיה לפי תחזית מזג האוויר ליום שבת. המשתמש יזין באופן ידני את זמני ההשקיה לפני כניסת השבת ולאחר שמירת הנתונים המערכת תעבוד באופן אוטומטי ביום שבת לפי הפרמטרים שאותם הזין המשתמש.

ניטור והצגת נתונים :

ניטור הנתונים יתבצע במהלך היום כל כמה שעות. יבוצע חישוב ממוצע עבור כל חיישן והנתונים יוצגו למשתמש בצורה גרפית (גרף בלוק או גרף "פאי")

ממש משתמש :

למשתמש תהיה אפליקציה ידידותית שבעזרתה הוא יוכל לראות את מצב הצמח לפי החיישנים, לקבוע מצבים, לראות נתונים גרפיים או טבלאיים ושליחת משוב כדי לעקוב בצורה יותר ברורה על ידי מתכנני המערכת לצורכי המשתמש.

אבטחה :

המערכת תהיה מאובטחת על ידי שימוש הצפנה וסיסמאות חזקות, יהיה מנגנון שיזהיר את המשתמש מפני סיסמא חלשה. עדכוני תוכנה קבועים כדי לתקן בעיות פרצות אבטחה ידועות. הימנעות בשימוש ספריות לא מהימנים, הצפנת הנתונים שנשלחים ומאוכסנים במערכת כדי למנוע גישה לא מורשית. שימוש בפרוטוקולי תקשורת מאובטחים וחומת אש כדי למנוע אבטחה לא מורשית. ביצוע גיבויים קבועים של הנתונים כדי שיהיה אפשר לשחזר אותם. שימוש בספק ענן אמין המציע שירותי אבטחה מתקדמים.

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

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

- חוסר תקשורת בהתנקות מהרשת

- חוסר תקשורת חיישנים
- ייצוג שגוי של ערכי החיישנים
- עומסי מתח העלולים להיווצר מעומס נתונים על השרת
- בעיות סנכרון בין חיישנים שונים

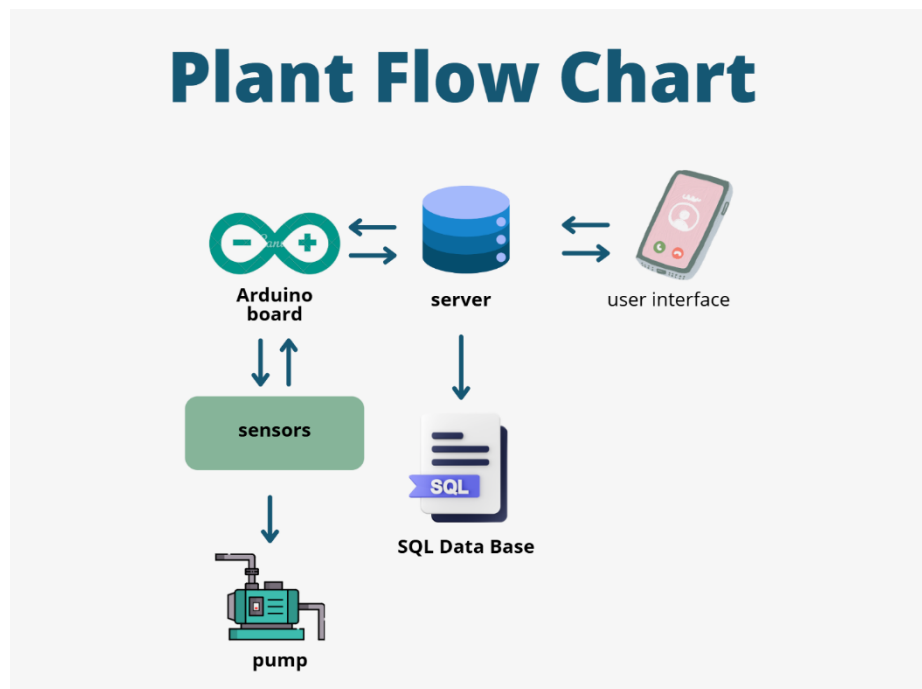
6.2 פתרונות אפשריים. (נא ציין פתרונות אפשריים וחלופות ארכיטקטוניות)

- התרעה בזמן אמת בהתנתקות מהרשת
- מנגנוני דגל המצביעים על חוסר תקשורת
- התראה בזמן שהחיישן מבצע קריאה או כמה קריאות לא נכונות
- העברת הנתונים אל מסמכי JSON והסרה של הנתונים המצטברים על השרת לאחר תקופה מסוימת
- יצירת מפתח עיקרי בו כל הטבלאות יהיו מסונכרנות לפי מפתח עיקרי וגם יצירת "זמן אחיד" בין שלל החיישנים

7. פתרון טכנולוגי נבחר:

7.1 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 וכו') פרויקט השקיית צמחים. המערכת מורכבת מלוח ארדואינו, חיישנים, שרת, ממשק משתמש בצד הלקוח, טווח תקשורת-אינטרנט, המערכת תיושם כאפליקציה לטלפון / טאבלט לשם נוחות המשתמש.



7.2 טכנולוגיות בשימוש. (איזה ומדוע בכמה מילים)

NodeJS – מפעיל את אפליקציית המשתמש לתקשורת בזמן אמת בין המשתמש למערכת המרכזית. הוא מספק ממשק ידידותי למשתמש להצגת נתוני חיישנים ומצבם, טיב איכות הצמחים, מערכת ניהול למצבי ההשקיה.

Arduino, - הלוח העיקרי האחראי על ניהול החיישנים ושליחה / קבלה של נתונים מהשרת ומחוצה לו

7.3 שפות הפיתוח: (איזה שפות ומדוע בכמה מילים?) שפת C, javascript, nodejs

Arduion – c++
Javascript – Front end && Back End

7.4. תיאור הארכיטקטורה הנבחרת- הסבר בכמה מילים מדוע

הארכיטקטורה הנבחרת היא IOT (מערכת מבוססת מבוססת) שימוש ב Arduino כלוח השולט על החיישנים שנמצאים ובקרה על השרת. עליו יהיו מותקנים החיישנים השולחים אל לוח ה Arduino את הנתונים המצביעים על טיב איכות הצמח. ממסד נתונים לאחסון הנתונים ושימוש ב NodeJS לתקשורת וממשק בזמן אמת. ארכיטקטורה זו מבטיחה זרימת נתונים יעילה אל המשתמש, ומאפשרת תזמון דינאמי וקבלת החלטות. בנוסף יכולת לשליטה על מצבים מסוימים כמו תמיכה בהשקיה אוטומטית בשבתות לאנשים דתיים, השקיה לפי מאפייני צמח, השקיה ידנית

7.5. חלוקה לתכניות ומודולים

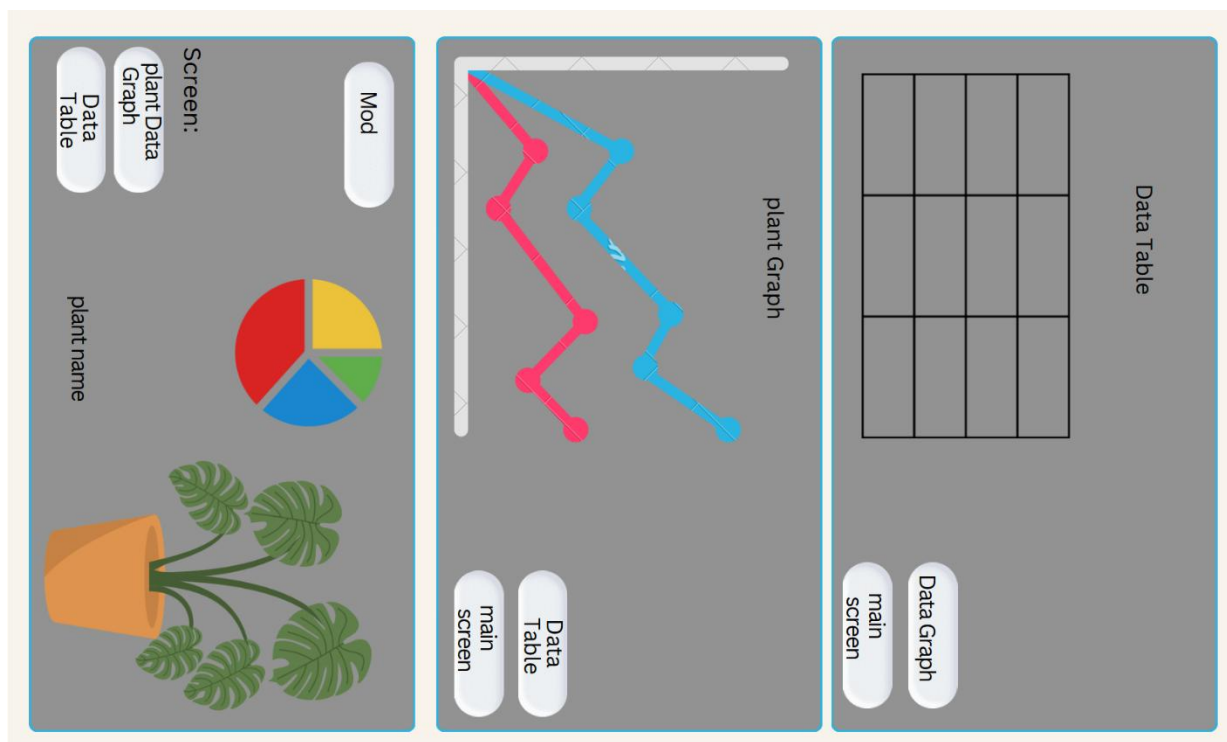
- מודל ביהוג החיישנים
מטפל בשליטה על החיישנים דרך קריאת נתונים מהשרת. כמו כן שולח נתונים מהחיישנים למשתמש כדי לנתר את לחות ההאדמה, טמפרטורה כמו כן קריאת מזג האוויר על ידי API כדי לקבוע את הזמן האופטימלי להשקיית הצמח.
- מודל ממשק משתמש
מספק ממשק ידידותי למשתמש מרובה צגים.
בצג הראשי יהיו נתוני הבריאות של הצמח וזמני ההשקיה העתידיים ומצב ההשקיה (אוטומטי, לפי סוג צמח, ידני או שבת).
במסכים הנלווים יהיו הנתונים הגרפיים והטבלאיים לפי דרישת הצמח המתבקש.
- מודל שרת
מנהל את אחסון והשליפה של הנתונים המגיעים והיוצאים מן לוח ה Arduino אל המשתמש
- מודל לקוח
מקבל את המידע מהשרת ומציג אותו אל הלקוח על ידי אפליקציה ייעודית .
כמו כן יוזם את התקשורת בין המשתמש לצמח בעזרת נתונים גרפיים וחווית משתמש.

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

ענן

סביבת העבודה הנבחרת היא סביבת ענן, המבטיחה תשתית עמידה בתקופת עומס

7.7. ממשק המשתמש/לקוח – GUI



7.8. ממשקים למערכות אחרות / API:

יש , קריאת נתוני מזג האוויר. המערכת תוכל להיות בבית ומחוץ לבית. קריאת מזג האוויר ההכרחית לצורך קביעת זמני ההשקיה . כאמור בימים חמים יותר האדמה תהיה פחות לחה ואור השמש יהיה חזק יותר ולהפך בימים קרים או גשומים.

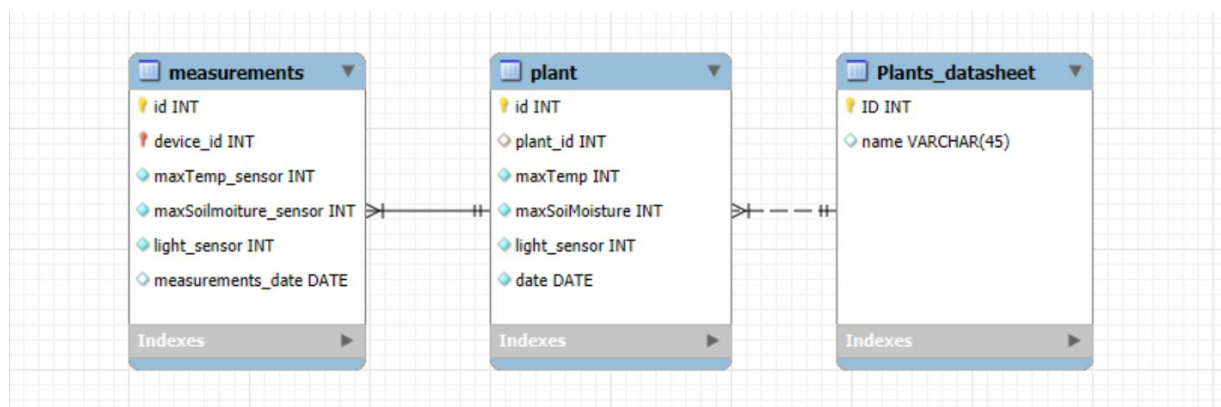
7.9. שימוש בחבילות תוכנה:

- Arduino IDE – לחומרת Arduino
- jetBrains – לבניית ממשק צד לקוח וצד שרת

8. שימוש במבני נתונים וארגון קבצים

8.1. נא פרט את מבני הנתונים.

עבור הארדוואינו הוא ישמור את חותמת הזמן , נתוני החיישנים , מצב בריאותם (עובד או לא עובד) ועיבוד התקשורת. עבור המשתמש הוא ישמור את הערכים האלה בטבלה ייעודית לכל קטגוריה והמידע יהיה זמין באפליקציה לאחר התחברות



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

אחסון מסד הנתונים יהיה ב MySQL , Arduino ישלח אל מסד הנתונים את מידע החיישנים כל כמה שעות. המידע ישמר ויאפשר ניתוח מעמיק יותר והבנה ברורה על ניהול המערכת

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

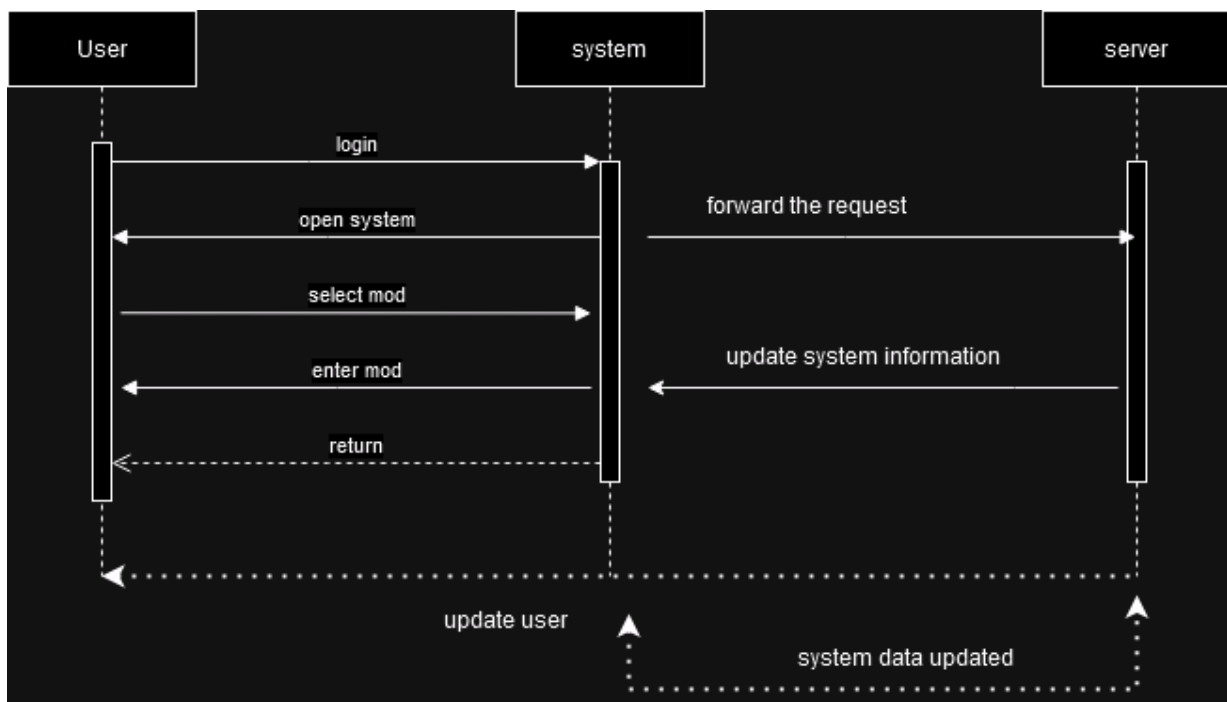
MySQL מבטיחה אינטגרציה בין הרכיבים הטבלאיים השונים על ידי יצירת מפתח משותף כמו כן חישוב נתונים ושמירתם במסד הנתונים עד לרגע קריסת המערכת במידה ותהיה . ברגע קריסה ה Arduino ימשיך את פעילותו באופן אוטומטי על פי הנתונים המאוחסנים מקומית ולפי התכנות הקודם. יחד עם זאת הפונקציונליות תהיה מוגבלת כמו למשל לא יהיה ניתן לעבור בין מצבים או לקבל התראה .

9. תרשימי מערכת מרכזיים

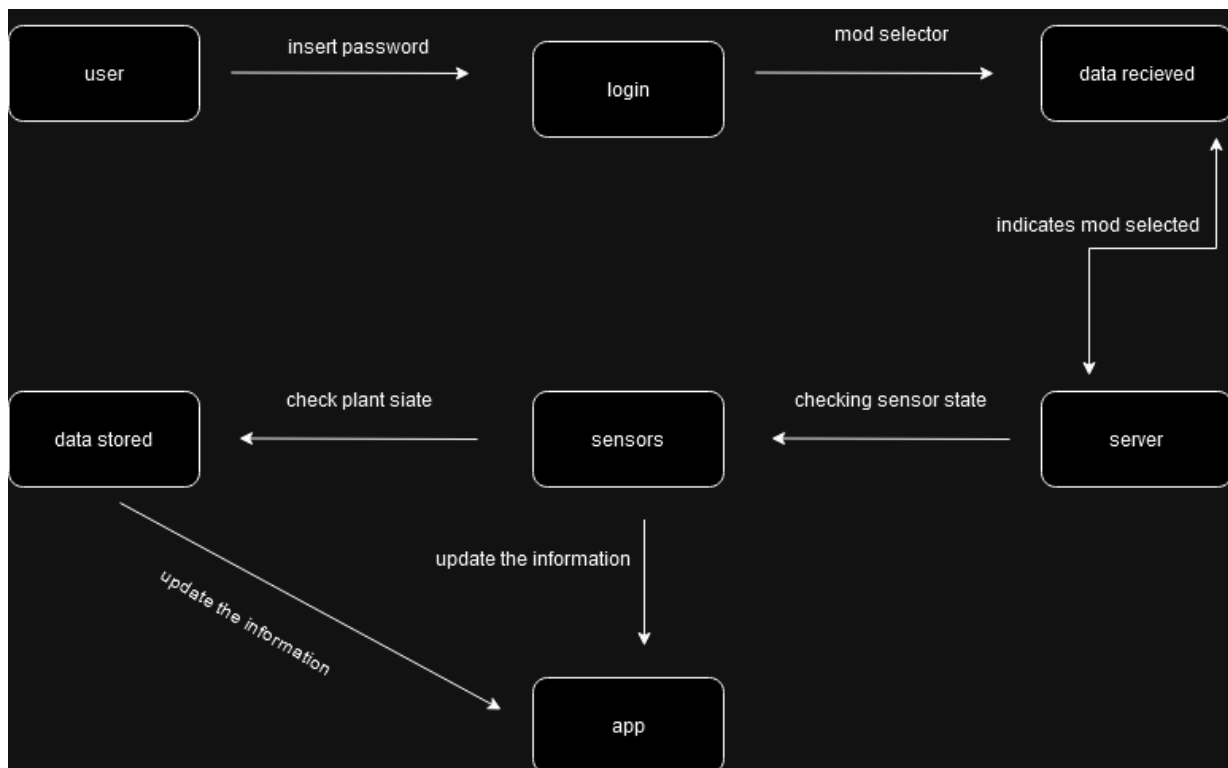
9.1 Use Case. אופן השימוש



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



Data flow 9.3



10. תיאור המרכיב האלגוריתמי – חישובי

10.1. איזה בעיה בא לפתור, איך יפתור?

אינו פותר בעיה

10.2. איסוף מידע וניתוחים סטטיסטיים (אנליטיקות)

מיושם איסוף נתונים אשר מציג מידע חיוני כמו כמות המים שהצמח קיבל , לחות אדמה והאור שאליו נחשף הצמח. שיפור קליטת נתונים מהחיישנים, עיבודם וניתוח המידע יאפשרו הבנה עמוקה יותר להמשך פעילויות בסביבות שונות ועם סוגי צמחים . בנוסף על כך יהיה ניתן לחזות בעזרת פרמטרים אלו עומסים על המערכת ואיך יהיה ניתן לחלק את העומסים / להוריד אותם על פי צורך.

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.

• שרת

איומים:

האקרים המנסים לשתול סקריפטים זדוניים (SQL INJECTION) , התקפות DDOS

תגובות:

הטמעת טכניקות סינון קלט והימנעות משימוש בתוכן דינמי לא מהימן (xss) . בחירת שמירת נתונים בענן מוכר ובעל מנגנוני הבטחה , שימוש בחומת אש למניעת התקפות DDOS וגיבוי קבוע של הנתונים

• אבטחת בקרת גישה לאתר:

איומים:

מניעת מפורץ לשנות את המידע בבסיס הנתונים

תגובות:

הגדרת הרשאות משתמשים, כך שכל משתמש יכול לגשת רק לנתונים הרלוונטיים לתפקידו. מנגנון אימות דו שלבי (סיסמא חזקה ותוקן למייל)

• אבטחת חשבונות משתמש:

איומים:

BRUTE FORCE שימוש בסיסמאות חלשות עלולות "לזמן" תוקף בקלות יתר אל המערכת.

תגובות:

אכיפת מדיניות סיסמה חזקה וארוכה (שילוב אותיות גדולות וקטנות, ספרות ותווים מיוחדים), הטמעת מנגנוני נעילת חשבונות לאחר מספר כניסות כושל, הטמעת אימות דו-שלבי וביצוע מעקב באופן קבוע אחר חשבונות משתמשים לאיתור פעילות חשודה

• אבטחת מסד נתונים:

איומים:

ווידוי שהמידע מהחיישנים מגיע מהמקום הנכון ולא מפורץ כלשהו

תגובות:

שימוש במפתחות הצפנה. לכל חיישן יהיה מפתח הצפנה ייחודי שיוודא שהתקשורת התקבלה מהמקור הנכון. ניתור אחר תנועות ברשת כדי לזהות פעילות חשודה כמו השקיה בזמן שלא הוגדר , עודף או חוסר בהשקיה

12. משאבים הנדרשים לפרויקט:

12.1.

מספר שעות המוקדש לפרויקט, חלוקת עבודה בין חברי הצוות

מספר השעות הנדרש - 300 שעות

12.2.

ציוד נדרש

- מחשב
- לוח ארדואינו
- חיישנים
- קיט להדמיית צמחים (מיכל אכסון עם אדמה ומיכל לאכסון מים)

12.3.

תוכנות נדרשות

- Arduino IDE
- JetBrains

12.4.

ידע חדש שנדרש ללמוד לצורך ביצוע הפרויקט

- שליחת תקשורת מהחיישנים אל השרת
- עיבוד המידע ויכולת ניהול מצבים

12.5. פורות ומקורות מידע

- <https://kits4.me/> - kits4me
- <https://www.arduino.cc/> - Arduino

13. תכנית עבודה ושלבים למימוש הפרויקט

- **Arduino** - התחלת כתיבת הקוד, הפיצ'ר הראשון יהיה הפעלה ידנית של ההשקיה מתוך עמוד אינטרנט
- **GUI** – עיצוב ופיתוח ממשק עבור המשתמש אשר יכלול את המיידע הרלוונטי עבורו
- **BackEnd** – כתיבת קוד צד שרת, הטמעת תקשורת בין המשתמש ללוח הארדואינו

14. תכנון הבדיקות שיבוצעו

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

מס בדיקה	בדיקה	תיאור בדיקה	תוצאה תקינה
1	ניסיון כניסה עם סיסמא שגויה מספר פעמים	המשתמש מנסה להיכנס למערכת עם סיסמא שגויה מספר פעמים	המערכת תחסום את המשתמש זמנית ותבקש ממנו לאפס את הסיסמא
2	בחירת מצב השקיה	המשתמש משנה את מצב ההשקיה מ"אוטומטי" ל"ידני" ואז חזרה ל"אוטומטי"	המערכת תשנה את מצב ההשקיה בהתאם להוראות המשתמש, ותמשיך להשקות את הצמח לפי לוח הזמנים החדש
3	שינוי הגדרות השקיה	המשתמש משנה את הגדרות ההשקיה כדי להתאים את מצב ההשקיה למצב הצמח ולמיקום שלו	המערכת מעדכנת את הנתונים בהתאם ותמשיך להשקות את הצמח לפי ההגדרות החדשות
4	ניטור נתוני חיישנים בזמן אמת	בדיקת תגובת המערכת כאשר יש תקלה בחיישן או במשאבה	המערכת מציגה באפליקציה התראה על תקלה בחיישן. מפסיקה את ההשקיה ועוברת למצב ידני.
5	קבלת התראות על תקלות או חריגות	יצירת תקלה מלאכותית בחיישנים כיסוי חיישן האור, חסימת צינור המשאבה, ניתוק כבל של אחד החיישנים, יצירת עליה או ירידה חדה בטמפרטורה.	אימות שהמשתמש מקבל התראה בזמן אמת דרך הודעה מהאפליקציה או דרך האימייל. סוג התקלה, הזמן שבו היא התרחשה, והחיישן הרלוונטי

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

(test

מס בדיקה	בדיקה	תיאור בדיקה	תוצאה תקינה
1	מודול לקריאה נכונה של נתונים מחיישנים בצורה נכונה	בדיקה שמטרתה לוודא שהמודול קורא נתונים מחיישנים בצורה נכונה וממיר אותם לפורמט המתאים לעיבוד נוסף. בדיקות כמו ערך חיובי עבור חיישן הלחות, טווח תקינות עבור חיישן הטמפרטורה וטיפול בשגיאת קריאה	המודול מחזיר ערך מספרי מדויק המתאים לקריאה בפועל של החיישן
2	מודול המאפשר למשתמש לשנות את הגדרות המערכת	בדיקה שמטרתה לוודא שהמשתמש יכול לשנות את הגדרות המערכת בצורה נכונה והשינויים נשמרים. בדיקה שהמשתמש יכול לשנות את זמני ההשקיה, להוסיף ולהסיר צמחים כמו כן בדיקה שהנתונים שהוזנו על ידי המשתמש אכן נשמרו במערכת גם אחרי סגירתה	הגדרות המערכת משתנות בהתאם לשינויים שבוצעו על ידי המשתמש
3	מודול המאפשר שחזור נתונים קודמים	בדיקה שמטרתה לוודא שהמודול יכול לשחזר נתונים קודמים בצורה נכונה ומלאה בדיקה שהמודול מחזיר בדיקה מיום מסוים, בדיקה שהמודול משחזר נתונים מטווח תאריכים מסוים ובדיקה שהמודול משחזר נתונים מסוג מסוים (למשל רק טמפרטורה)	המודול מחזיר את הנתונים ששומרו בעבר בצורה מדויקת

15. בקרת גרסאות (version control)

<https://github.com/Dorelhayani/HydroBuddy>

חתימת המנחה האישי

חתימת הסטודנט

ב. הערות ראש המגמה במכללה

ג. אישור ראש המגמה

שם: _____ חתימה: _____ תאריך: _____

ד. הערות הגורם המקצועי מטעם מה"ט

ה. 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 העסקי של השרת, בחירה בשפות עיליות JAVA, C#, או nodeJS.
 - 2.1.2. לתכולה חיונית/אלגוריתמית- מימוש בשפת native נניח C, C++.
 - 2.1.3. לצד לקוח AngularJS, Asp.net 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. ניתן להשתמש גם במסגרת frameworks קיימים כדוגמת dotNet.

4. מרכיב אלגוריתמי/חשובי-

- ישנה חשיבות רבה להתנסות של התלמיד והתמודדות עם יכולות חיוניות במסגרת מכלולי הפרויקט. ניתן לשלב היבטים אלגוריתמיים או לחילופין ניתוחים וחינוכים סטטיסטיים בסיסיים מעל מאגר נתונים, למשל:
- 4.1. במקרה של אתרים כניסת משתמשים, גיאומטריה חיונית 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\התמודדות עם עומסים.
 - 16.3. דרישות פונקציונאליות.
 - רשימת דרישות המשתמש מהמערכת, מהן הפעולות בהן נדרשת המערכת לתמוך.
- 5.2. חלופות ארכיטקטוניות-
 - 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) 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###, http, https, ssl###, 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