***4.3.25***

**מסמך מסכם לפרויקט גמר   
בהתמחות פיתוח מערכות תוכנה  
*במחלקה למדעי המחשב והמידע***

הפקולטה להנדסה, המרכז האקדמי רופין

מוגש ע"י:

**אושר גרופי - 205762586**

**מיכאל גרינברג - 208570572**

**רועי טרבינוביץ - 316225382**

מנחה/י הפרויקט:

בני בורנפלד

**הערות כלליות (**מיועדות לגברים ונשים כאחד**):**

1. מסמך מסכם לפרויקט הגמר מתעד את כל המידע הנדרש לפיתוח מערכת תכנה ואת כל שלבי העבודה. פיתוח מערכת תכנה מתייחס בעיקר לפיתוח מיזמים מבוססי אפליקציה, ולתוכנות שיכולות לשמש כחלק מפרויקט גדול יותר, או כיחידת stand alone. בכל המקרים – המונח שעושים בו שימוש לאורך המסמך הוא "**מערכת**".
2. בתבנית המצורפת לשימושכם, קיימים 10 סעיפים, שיסייעו לכם להתקדם בפרויקט באופן מובנה, ולתעד את שלבי העבודה במהלך פיתוח מערכת התכנה שלכם (בקיצור, "מערכת").
3. יתכן וחלק מהסעיפים אינם רלוונטיים לפרויקט שלכם. במקרה כזה, יש להתייעץ עם המנחה ולאחר אישורו, ניתן לא להתייחס לסעיפים אלו במסמך. **אין למחוק אותם** כדי לא לשבש את המספור. במקום, יש להוסיף הערה בתחילת הסעיף: "סעיף זה אינו רלוונטי לפרויקט זה".
4. על פי הצורך, ניתן להוסיף סעיפים שלא מופיעים בתבנית המצורפת.
5. תרשימים שאתם עושים בהם שימוש לכל המסמך – ככל שניתן, יש להשתמש בתרשימי UML.
6. אתם יכולים לעשות שימוש בכלי AI. בעת שימוש כזה, יש לפרט את האופן, את התוצרים במידה ויש, ואת הבקרה שעשיתם על תוצרים אלו.
7. לאורך המסמך מופיעות הערות אדומות שנועדו לספק לכותבי המסמך מידע נוסף שיסייע להם לכתוב את המסמך על הצד הטוב ביותר. **יש להורידם מהמסמך לאחר סיומו.**

בהצלחה!

# תקציר מנהלים

סעיף זה מספק תקציר על הפרויקט, ולכן יש לכתוב אותו לאחר השלמת כל יתר הסעיפים במסמך. על התקציר להתייחס לכלל סעיפי המסמך. התעכבו על הבעיה העיקרית '/או הצורך העיקרי עימם מתמודדת המערכת שלכם. הסבירו מהי המוטיבציה לפיתוח המערכת, כלומר, מדוע וכיצד המערכת המוצעת תפתור את הבעיה/צורך הקיימ/ת. מנו את המטרות העיקריות של המערכת, ותארו כיצד היא תתמוך בפונקציונליות של משתמשיה.

## תקציר מנהלים

## הגורם המבצע

* רשמו את פרטי חברי הצוות שביצע את הפרויקט (פרטי סטודנטים)

אושר גרופי - 205762586

מיכאל גרינברג - 208570572

רועי טרבינוביץ - 316225382

* הוסיפו את פרטי המנחים האקדמיים, המנחים בתעשייה (אם יש), פרטי הארגון ואנשי הקשר (אם התבצע בארגון), ועוד בעלי עניין לפי ראות עיניכם.  
    
  בני בורנפלד

# הצגת הבעיה/הצורך

**תלות במספר שחקנים** – המשחק דורש לפחות ארבעה משתתפים (שני מפעילים ושני סוכנים), מה שמקשה על זמינות המשחק. ללא מספר שחקנים מינימלי, המשחק לא יכול להתקיים, דבר שעלול למנוע מאנשים לשחק בכל רגע שירצו.  
**אתגר במתן רמזים איכותיים** – יצירת רמזים חכמים ומדויקים היא משימה לא פשוטה עבור שחקנים רבים. חוסר יצירתיות או חשש מרמזים לא טובים מפחיתים את איכות החוויה ועלולים לפגוע בהנאה מהמשחק.

המטרה היא **להפוך את המשחק לנגיש וכיף יותר**, גם במקרים שבהם אין מספיק שחקנים אנושיים ע"י שילוב AI כמפעיל או סוכן וכך לשחק תמיד וליצור חוויית משחק חכמה ומאתגרת יותר בעזרת כלים מתקדמים.

# רקע לבעיה: תיאור השוק/הארגון והסביבה העסקית

## **תיאור** השוק/הארגון והצורך במערכת

**אופי השוק/הארגון והסביבה העסקית**

* היסטוריה קצרה של השוק/הארגון והשחקנים המרכזיים בשוק (שאינם מתחרים)

Czech Games Edition המוציא לאור המקורי של המשחק

הארגון שפיתח את המשחק המקורי והפיץ אותו בשוק העולמי.

מתמקד בעיקר במשחקי לוח פיזיים, אך מציע גם גרסה דיגיטלית מוגבלת שאינה תחרות ישירה.

* אופי התעשייה ומדדים חשובים בענף המודדים את הצלחת העוסקים בענף (למשל: היקף המכירות, מדדי שביעות רצון, כמות הלקוחות החוזרים, אחוזי צמיחה, אחוזי הנטישה ועוד)  
   **אופי התעשייה**

תעשיית משחקי הלוח והמשחקים הדיגיטליים היא תעשייה משולבת, המחברת בין חוויות משחק מסורתיות (פיזיות) לבין פלטפורמות דיגיטליות המאפשרות משחק אונליין מרובה משתתפים. בשנים האחרונות נרשמה צמיחה משמעותית בקטגוריית משחקי הלוח הדיגיטליים, במיוחד בעקבות מגפת הקורונה שהגבירה את הצורך במשחקים שניתן לשחק מרחוק.

תעשיית משחקי הלוח והמשחקים הדיגיטליים מחולקת למספר תתי-ענפים עיקריים:

**משחקי לוח פיזיים**.

**גרסאות דיגיטליות של משחקי לוח**.

**משחקים מבוססי בינה מלאכותית.**

כיום, משחקים דיגיטליים עם אלמנטים של עיבוד שפה טבעית (NLP) ובינה מלאכותית הופכים לפופולריים יותר, מאחר והם מאפשרים משחקים אינטראקטיביים גם במצב של שחקן יחיד, או כאשר חסרים משתתפים. הפרויקט שלנו משתלב בתחום זה ומציע פתרון חדשני בשוק.

**מדדים חשובים בענף**

היקף המכירות הגלובלי של משחקי לוח דיגיטליים  
בשנת 2023, שוק משחקי הלוח (פיזי ודיגיטלי) הוערך בכ-17.3 מיליארד דולר, עם תחזית צמיחה שנתית של 7.2% עד שנת 2028.  
פלח השוק הדיגיטלי צומח במהירות גבוהה יותר, כאשר גרסאות אונליין למשחקים פופולריים זוכות להצלחה משמעותית.

* תקנונים ואילוצים ענפיים ורגולטוריים שיש להתחשב בהם (אם קיימים)

**פרטיות והגנה על נתונים**

לקבל אישור מהמשתמשים לפני שמאחסנים את הנתונים שלהם.

לדאוג לאבטחה כדי למנוע דליפות מידע ופריצות.

**תקנות לשימוש ב-AI**  
המשחק משתמש בבינה מלאכותית כדי לייצר רמזים, אז חשוב לוודא:

שה-AI לא מייצר רמזים לא הוגנים או בעייתיים.

שהוא לא מפר זכויות יוצרים (למשל, אם משתמשים במודלים חיצוניים).

שהמשתמשים יודעים שהם משחקים מול AI ולא מול אדם.

**אבטחת מידע**  
כדי למנוע פריצות או דליפות מידע נבצע הגבלת גישה לנתונים רק למשתמשים מורשים.

**מאפייני קהל היעד של לקוחות ומשתמשי המערכת**

* מהו האפיון הדמוגרפי של הלקוחות הפוטנציאליים של המערכת? במידה וקיימים משתמשים שונים, יש לבצע אפיון עבור כל פלח (סגמנט).

המשחק מתאים **לכל הגילאים**, מאחר שהוא מבוסס על חשיבה אסוציאטיבית ושיתוף פעולה, ולא דורש מיומנויות טכניות מסובכות.

**ילדים ובני נוער** – יכולים ליהנות מהמשחק כחוויה חברתית וכדרך לפיתוח חשיבה מילולית.

**מבוגרים וצעירים** – משחק אסטרטגי מהנה שיכול לשמש לבילוי עם חברים או משפחה.

**שחקנים מזדמנים** – אנשים שמחפשים משחק קליל עם אפשרות למשחק מרחוק.

**שחקנים תחרותיים** – שחקנים שמעוניינים לפתח אסטרטגיות ולשחק עם חברים ברמה גבוהה יותר.

בזכות האפשרות לשחק **מול AI** או עם חברים אונליין, המשחק נגיש לכולם, כולל שחקנים שאין להם צוות מלא או שמחפשים חוויית משחק מאתגרת גם לבד

* מהן התכונות של הלקוחות שהופכות אותם ללקוחות המערכת (יש להבחין בין לקוחות קיימים – המשתמשים במוצר/שרות קיים שאינו שלכם, ולקוחות עתידיים שלא עושים כל שימוש במוצר/שרות דומה, ושבעקבות פיתוח המערכת - יצטרפו למעגל הלקוחות שלכם).

**תכונות הלקוחות שהופכות אותם ללקוחות המערכת**

**לקוחות קיימים** – משתמשים שכבר משחקים במשחקי מילים/אסטרטגיה

**מי הם?**

אנשים שכבר משחקים בגרסה הפיזית של "שם קוד" או בגרסה הדיגיטלית הקיימת.

שחקני משחקי מילים ואסוציאציות אחרים כמו "Wordle", "Scrabble" ו-"Taboo".

חובבי משחקי אסטרטגיה קבוצתיים שמחפשים משחקים מבוססי חשיבה ולוגיקה.

למה הם יצטרפו למערכת שלנו?

אפשרות לשחק מרחוק עם חברים ולא רק פנים אל פנים.

שילוב AI מתקדם שמספק רמזים – משדרג את החוויה ומאפשר משחק יחיד.

תמיכה במצבי משחק שונים – AI כמפעיל, משחק תחרותי או מצב לימודי.

שיפור חוויית המשחק – מערכת חכמה שמנהלת את הלוח, החוקים והתור באופן אוטומטי.

**לקוחות חדשים** - משתמשים שעד היום לא שיחקו במשחקי מילים

**מי הם?**

אנשים שאוהבים משחקים חברתיים, אבל מעדיפים משחקים דיגיטליים על פני משחקי קופסה.

שחקנים יחידים שאין להם פרטנרים קבועים ויחפשו משחק שיכול להתנהל גם נגד AI.

מורים ואנשי חינוך שמעוניינים לשלב את המשחק ככלי לימודי לכיתות או לאימוני חשיבה אסוציאטיבית.

אנשים סקרנים לגבי בינה מלאכותית ורוצים לראות איך AI מתמודד עם משחקי מילים ואסוציאציות.

**למה הם יצטרפו למערכת שלנו?**יכולת לשחק בלי 4 שחקנים אנושיים והכנסת AI כשחקן נוסף.חדשנות טכנולוגית – שילוב AI שמציע רמזים ומאפשר משחק אסטרטגי מולו.

נגישות אונליין – אפשרות לשחק מכל מקום וללא צורך במפגש פיזי.

ללא צורך בהיכרות מוקדמת עם משחקי מילים – המערכת מלווה את השחקן ונותנת משוב.

פוטנציאל ללמידה ופיתוח חשיבה יצירתית – מתאים גם למטרות חינוכיות ואימוני מוח.

**פתרונות חלופיים קיימים**

* תארו פתרונות חלופיים (חלופות) קיימים שיכולים לספק מענה למערכת הנדרשת. החלופות יכולות להיות מערכות אחרות, ארכיטקטורות, טכנולוגיות, ומאמרים שהופיעו בספרות אקדמית/ מקצועית.
* יש לפרט על החלופות השונות באמצעות טבלה 1.

| **שם החלופה** | **תיאור החלופה** | **יתרונות ותועלות** | **חסרונות וחולשות** | **לינק למקור מידע** | **הערות** |
| --- | --- | --- | --- | --- | --- |
| **מודלים של ai** |  |  |  |  |  |
| **שימוש במודלים של NLP כמו OpenAI GPT ליצירת רמזים** | **שימוש במודל בינה מלאכותית לניתוח מילים והפקת רמזים אסוציאטיביים בצורה חכמה.** | **- יכולת לנתח מילים וליצור רמזים חכמים.  - מאפשר משחק מול AI גם כשאין מספיק שחקנים אנושיים.** | **- עשוי לתת רמזים לא ברורים או מוטים.  - דורש התאמות כדי למנוע תשובות שגויות או לא תואמות למשחק.** | **https://chatgpt.com/** | **ניתן לשלב אך דורש כיול של המערכת.** |
| **Google Bard** | **מודל שפה גדול מבית גוגל, מבוסס על PaLM 2, מאפשר דיאלוגים אינטראקטיביים.** | **- שילוב קל עם שירותי Google Cloud.**  **- מבוסס על מידע עדכני מהאינטרנט.**  **- ניתן לשלב עם Firebase ו-Dialogflow.** | **- דורש חיבור רציף לאינטרנט.**  **- הגבלת API עלולה להיות בעייתית לפרויקטים גדולים.** | **https://bard.google.com/** | **מתאים לאינטגרציה עם Google Cloud.** |
| **Databases:** |  |  |  |  |  |
| **SQL Server** | **מסד נתונים יחסי (RDBMS) מבית Microsoft, המציע יכולות מתקדמות לניהול נתונים, כולל תמיכה בטרנזקציות, אבטחה, ויכולת סקלאביליות גבוהה.** | **- אינטגרציה מושלמת עם ASP.NET ו-C#.**  **- תמיכה מובנית בטרנזקציות וניהול נתונים מורכב.**  **- שירותים מובנים כמו SQL Server Management Studio (SSMS) ו-Azure SQL.** | **- רישוי מסחרי (עשוי להיות יקר לעומת פתרונות קוד פתוח כמו PostgreSQL).**  **- מורכבות גבוהה יותר בניהול שרתים מאשר Firebase או NoSQL.** | **https://www.microsoft.com/en-us/sql-server/** | **מתאים למערכות מבוססות .NET, ולאפליקציות שדורשות סקלאביליות ואבטחה חזקה.** |
| **MongoDB Atlas** | **מסד נתונים NoSQL מבוזר בענן, מתאים למשחקים ומבוסס מסמכים (JSON).** | **- ביצועים גבוהים בניהול נתוני משתמשים.**  **- תמיכה מובנית באינדקסים דינמיים.**  **- גישה מהירה לנתונים.** | **- דורש ידע בתפעול NoSQL.**  **- ייתכן שיהיו עלויות גבוהות בפרויקטים גדולים.** | **https://www.mongodb.com/products/platform/atlas-database** | **מתאים לאפליקציות מבוזרות עם הרבה משתמשים.** |
| **צד לקוח** |  |  |  |  |  |
| **React.js** | **ספריית JavaScript פופולרית לפיתוח ממשקי משתמש אינטראקטיביים.** | **- מבנה מודולרי ומאפשר שימוש חוזר בקומפוננטים.**  **- ביצועים גבוהים בזכות Virtual DOM.** | **- דורש ידע טוב ב-JavaScript.**  **- עשוי להיות מורכב יותר ממשקים פשוטים יותר.** | **https://react.dev/** | **מתאים לפיתוח אפליקציות משחקים מבוססות דפדפן.** |
| **Angular** | **פריימוורק לפיתוח Single Page Applications (SPA) מבית Google, תומך ב-TypeScript ומאפשר בניית אפליקציות דינמיות.** | **- מבנה מבוסס מודולים שמקל על ניהול קוד.**  **- מתאים לפרויקטים גדולים עם דרישות מורכבות.**  **- תומך ב-Routing ו-State Management בצורה אינטגרלית.** | **- כבד יותר בהשוואה ל-React ו-Vue.**  **- ההתחלה עשויה להיות מורכבת יותר למפתחי פרונט-אנד חדשים.** | **https://angular.dev/** | **פתרון חזק לפרויקטים גדולים שדורשים מניפולציות מורכבות של נתונים.** |
| **צד שרת** |  |  |  |  |  |
| **ASP.NET Core** | **מסגרת פיתוח Cross-Platform לניהול שרתים מבוססי C#, תומך ב-REST API, WebSockets, ו-Entity Framework לניהול נתונים.** | **- ביצועים גבוהים (Fast & Scalable).**  **- תמיכה מעולה במסדי נתונים SQL (כגון SQL Server, PostgreSQL).**  **- מתאים לאפליקציות Enterprise.** | **- דורש ידע ב-C# ובניהול IIS/Kestrel.**  **- פחות קהילתי מפתרונות מבוססי Node.js.** | **https://dotnet.microsoft.com/en-us/apps/aspnet** | **מתאים במיוחד אם המשחק שלך כבר מבוסס .NET ו-SQL Server.** |
| **Node.js + Express.js** | **סביבת JavaScript לניהול צד שרת עם תמיכה ב-REST API, WebSockets ו-NoSQL.** | **- קל ופשוט ליישום.**  **- תומך ב-WebSockets לניהול משחקים בזמן אמת.**  **- אינטגרציה טובה עם מסדי נתונים NoSQL (MongoDB, Firebase, Redis).** | **- ביצועים נמוכים יחסית ביישומים CPU-Intensive.**  **- פחות מותאם ליישומים מבוססי SQL Server.** | **https://expressjs.com/** | **מתאים למשחקים מהירים ודינמיים, במיוחד אם יש שימוש ב-WebSockets.** |
| **ניהול משחק בזמן אמת בין שחקנים** |  |  |  |  |  |
| **שימוש ב- Firebase לניהול חדרים ושחקנים** | **טכנולוגיה לניהול חדרים, שחקנים ועדכון נתונים בזמן אמת.** | **- תומך במשחקים מרובי משתתפים אונליין.  - מאפשר עדכון מיידי של לוח המשחק והמשתתפים.** | **- לא כולל באופן מובנה תמיכה בבינה מלאכותית.  - דורש חיבור למערכת צד שרת לניהול משחקים.** | **https://firebase.google.com/** | **מתאים לניהול משחקים אונליין, אך לא כתחליף לAI .** |
| **Supabase** | **פלטפורמת קוד פתוח לניהול מסדי נתונים (PostgreSQL) עם תמיכה ב-WebSockets וניהול נתונים בזמן אמת.** | **- תמיכה מובנית ב-WebSockets (Realtime API).**  **- ניהול אימות משתמשים (Auth) כמו ב-Firebase.**  **- תמיכה ב-PostgreSQL עם פונקציונליות דומה ל-Firestore.** | **- פחות אופטימלי לניהול אירועים בקנה מידה רחב (לא כמו Firebase Realtime DB).**  **- דורש ניהול שרתים אם משתמשים בגרסה הפרטית.** | **https://supabase.com/** | **תומך בWebSockets, מתאים לניהול תורות בין שחקנים.** |

*טבלה 1. חלופות*

סיכום קצר לגבי ממצאים שעלו מן החלופות

ASP.NET Core - צד שרת

-פתרון חזק ומבוסס לפיתוח Backend עם תמיכה מצוינת ב-SQL Server

-מספק ביצועים גבוהים, אבטחה וסקלאביליות

-מתאים לפרויקטים גדולים ומורכבים, אך דורש ידע מוקדם בניהול שרתים ופיתוח ב-C#

React.js - צד לקוח

-פריימוורק גמיש ומהיר לפיתוח ממשקי משתמש אינטראקטיביים

-קהילה רחבה, אקוסיסטם עשיר וספריות רבות שמאפשרות התאמה אישית

-דורש ניהול State חכם כמו Redux או Context API לפרויקטים גדולים

SQL Server - אחסון נתונים יחסיים

-מסד נתונים אמין ויציב, המתאים למערכות גדולות הדורשות אבטחה וטרנזקציות מורכבות

-אינטגרציה מצוינת עם ASP.NET, כולל תמיכה בענן דרך Azure SQL

-דורש רישוי מסחרי, מה שעלול להיות יקר יותר בהשוואה לחלופות קוד פתוח כמו PostgreSQL

Firebase - אחסון וניהול נתונים בזמן אמת

-פתרון פשוט ומהיר לניהול נתונים בזמן אמת, מתאים למשחקים מרובי משתתפים

-לא דורש ניהול שרתים, מספק גם Authentication ו-Hosting מובנים

-מגבלות עיקריות הן תלות ב-Google, סקלאביליות יקרה, וגישה פחות גמישה לנתונים מאשר SQL Server

GPT NLP - עיבוד שפה טבעית למשחקים

-מעולה ליצירת דיאלוגים דינמיים במשחקים, מאפשר אינטראקציות חכמות עם שחקנים

-ניתן לשלב עם Firebase או ASP.NET כדי לנהל צ'אט חכם ואינטראקטיבי

-דורש גישה רציפה לרשת, עלול להיות יקר בשימוש קבוע ב-API

## Pains and Gains

* Pains - מהן הבעיות הקיימות כיום אצל הלקוחות העתידיים של המערכת?

**1. קושי לשחק כשאין מספיק שחקנים**  
משחק "שם קוד" דורש לפחות ארבעה שחקנים, מה שמגביל את היכולת לשחק בזמנים שונים או בקבוצות קטנות. שחקנים רבים נתקלים בקושי למצוא מספיק משתתפים, במיוחד אם רוצים לשחק באופן מקוון.

**2. קושי במתן רמזים איכותיים**  
המפעיל (Spymaster) חייב לתת רמזים חכמים ומדויקים, אך לא לכל השחקנים יש את היכולת האסוציאטיבית הנדרשת.

שחקנים חסרי ניסיון מתקשים לספק רמזים איכותיים.

לעיתים רמזים שניתנים מבלבלים את הקבוצה, מה שמוביל למשחק פחות מהנה.

**3. מגבלות כמות אנשים במשחק**

קיימות גרסאות מקוונות של "שם קוד", אך הן מבוססות על שחקנים אנושיים בלבד.

אין מנגנון AI שיכול להחליף שחקנים חסרים, מה שמגביל את המשחק.

אין תמיכה בניהול תורות חכם ובשיפורי משחק אינטראקטיביים.

* Gains - מה המערכת העתידית יספק ללקוחות שלא קיים היום?

**1. אפשרות לשחק גם בלי ארבעה שחקנים**

המערכת תאפשר לשחק גם כשאין מספיק משתתפים, על ידי שילוב AI כמפעיל (Spymaster) אוכסוכן (Agent).

שחקנים יוכלו לשחק אחד נגד השני, בזוגות או אפילו לבד מול AI.

**2. AI חכם שמספק רמזים ולומד מהמשחק**

ה-AI יספק רמזים מתוחכמים ואסוציאטיביים, בדומה לשחקן אנושי מיומן.

עם הזמן, ה-AI ילמד ויתאים את עצמו לסגנון המשחק של המשתתפים, מה שיאפשר אתגר ועניין מתמשך.

אם שחקן מתקשה במתן רמזים, ה-AI יוכל להציע עזרה ולתת דוגמאות.

## מטרות ומדדים

תארו את מטרות המערכות

**1. יצירת חוויית משחק דיגיטלית נגישה ומתקדמת**

לאפשר משחק מקוון מלא ללא צורך במשחק פיזי.

לספק חוויית משחק חלקה ומסונכרנת לכל המשתתפים.

להפוך את המשחק לזמין בכל זמן, ללא תלות בשחקנים אחרים.

**2. שילוב AI חכם לניהול המשחק**

לפתח AI שמסוגל לתפקד כמפעיל (Spymaster) ולתת רמזים חכמים לשחקנים.

ליצור AI שמסוגל לנחש מילים בצורה לוגית ולשחק בתור סוכן (Agent).

להציע עזרה לשחקנים חדשים על ידי מתן רמזים או הצעות לשיפור הניחושים.

**3. פתרון בעיית המחסור בשחקנים**

לאפשר משחק בכל הרכב אפשרי, כולל משחק יחידני נגד AI.

לאפשר משחק בזוגות, בקבוצות קטנות או במשחקים פתוחים עם אנשים אונליין.

**4. פיתוח מערכת מאובטחת ויעילה לניהול נתונים בזמן אמת**

להשתמש ב-Firebase לניהול חדרי המשחק, השחקנים והעדכונים בזמן אמת.

להבטיח שימוש בטוח ומוגן בעזרת מנגנוני פרטיות ואבטחת מידע.

לתמוך בנגישות לכל סוגי המשתמשים, כולל תמיכה בצבעים נגישים, שליטה נוחה וללא מגבלות פיזיות.

* ציינו את מדדי ההצלחה הכמותיים של המערכת העתידית וכיצד ניתן למדוד אותם

1. אחוזי דיוק של ה-AI ברמזים ובניחושים. יש למדוד את אחוז הרמזים שה-AI סיפק ונבחרו על ידי השחקנים כהגיוניים ואת אחוז הניחושים שה-AI ביצע ופגע במילים הנכונות של הקבוצה. המדידה תתבצע באמצעות מעקב אחר הצלחת הרמזים באמצעות דירוג של השחקנים (האם הרמז עזר להם) והשוואת אחוזי הניחושים המוצלחים לעומת ניחושים שפגעו במילים נייטרליות או של היריב.
2. אחוזי שביעות רצון של המשתמשים מרמזי ה-AI. יש למדוד את אחוז השחקנים שמדרגים את הרמזים של ה-AI כרלוונטיים ואיכותיים ואת אחוז השחקנים שחוזרים לשחק וממשיכים להשתמש ב-AI. המדידה תתבצע על ידי הוספת כפתור דירוג אחרי כל רמז של AI ("האם הרמז היה מועיל?") ומעקב אחר שימוש חוזר ב-AI במשחקים עוקבים.
3. אחוז השחקנים המעדיפים משחק עם AI. יש למדוד כמה משתמשים בוחרים לשחק עם AI אפילו כשיש להם אפשרות לשחק עם בני אדם והאם שחקנים מעדיפים AI כמפעיל או כסוכן. המדידה תתבצע על ידי השוואה בין משחקים עם AI בלבד לבין משחקים שבהם רק שחקנים אנושיים משתתפים וביצוע סקר שחקנים בנוגע להעדפה שלהם בין AI לשחקנים אנושיים.
4. אחוזי ההצלחה של AI לעומת שחקנים אנושיים. יש למדוד האם AI נותן רמזים טובים יותר משחקנים אנושיים והאם AI מנחש מילים בצורה מדויקת יותר מאשר שחקנים רגילים. המדידה תתבצע על ידי השוואת אחוזי ההצלחה של AI מול שחקנים אמיתיים בעזרת ניתוח נתונים מצטברים ובדיקת מי משיג יותר נקודות – AI או שחקנים אמיתיים.

## שיטות לאיסוף נתונים

סעיף זה אינו רלוונטי לפרויקט זה

# תיאור ראשוני של הפתרון המוצע והאתגרים שבדרך

המערכת שנבנית היא משחק מקוון של "שם קוד" בשילוב בינה מלאכותית (AI).

הפתרון מאפשר משחק מרובה משתתפים בזמן אמת, עם אפשרות לשלב שחקנים אנושיים ולשחק גם מול AI. המערכת תתבסס על Firebase לניהול המשחק, תורי המשחק ועדכון מצב הלוח בזמן אמת, ותשתמש במודל OpenAI(GPT) ליצירת רמזים חכמים.

**המערכת תאפשר למשתמשים:**

* להתחבר וליצור חשבון אישי.
* להזמין חברים או לשחק עם AI.
* לבחור תפקיד במשחק (סוכן/מפעיל).
* לשחק עם לוח רנדומלי שנוצר עבור כל משחק.
* לקבל רמזים חכמים מה-AI או ליצור רמזים בעצמם.
* לצפות בביצועים לאחר כל משחק.

**משתמשי המערכת והפונקציונליות של כל אחד מהם:**

**שחקן אנושי-**

* יוכל להתחבר למערכת וליצור משחק או להצטרף למשחק קיים.
* יוכל לבחור תפקיד סוכן (Agent) או מפעיל (Spymaster).
* אם משחק כסוכן – יקבל רמזים מהמפעיל וינסה לזהות את המילים הנכונות.
* אם משחק כמפעיל – ייתן רמזים לחברי קבוצתו בהתבסס על המילים שעל הלוח.

**בינה מלאכותית (AI)-**

* יוכל לתפקד כמפעיל ולספק רמזים מבוססי NLP.
* יוכל לעזור לסוכנים על ידי מתן משוב על בחירותיהם.

**מנהל המשחק (Firebase – Backend)-**

* יוודא שהמשחק מתנהל לפי הכללים.
* ישמור את מצב המשחק ויעדכן אותו בזמן אמת לכל המשתמשים.
* ינהל את תורי השחקנים וישלח התראות במידת הצורך.
* יסנכרן את המידע בין כל השחקנים באמצעות Firebase Realtime Database.

**מידע עיקרי שישמר במערכת-**

* נתוני משתמשים (שם משתמש, היסטוריית משחקים, חברים).
* יאחסן נתוני משחק היסטוריים לניתוח שיפור חוויית המשחק.
* מצב המשחקים בזמן אמת (מיקום מילים על הלוח, תורים פעילים, ניקוד).
* סטטיסטיקות (אחוז הצלחה בזיהוי מילים, ביצועי AI מול שחקנים).

**אתגרים אלגוריתמיים וצפויים בדרך-**

* **ניהול סנכרון בזמן אמת –** הבטחת תקשורת תקינה בין כל השחקנים כך שהלוח, התור והמשוב יתעדכנו ללא השהיות.
* **יצירת חווית משתמש אינטואיטיבית –** תכנון ממשק משתמש ברור ונוח עם התראות על התור, פעולות נדרשות ומשוב על ביצועים.

# תכנית עבודה של הפרויקט

**לוח זמנים מתוכנן להמשך פיתוח –**

**שלב 1 – פיתוח בסיסי והקמת תשתית (1.4 - 21.4)**

* יצירת חיבור בין מסכי המשחק והוספת ניווט בין הדפים.
* חיבור Firebase לניהול משתמשים, יצירת חדרים והגדרת תפקידים.
* פיתוח לוגיקת משחק בסיסית (ניהול תורות, שיוך מילים לקבוצות).

**שלב 2 – שילוב ה-AI והתאמה ראשונית (22.4 - 12.5)**

* הטמעת מודל OpenAI GPT4 לצורך יצירת רמזים ראשוניים.
* בדיקות ראשוניות לאיכות הרמזים ושיפורים על ידי פרומפטים מדוייקים.
* יצירת מנגנון של הצגת רמזים למשתמשים וניהול תורות על בסיסם.

**שלב 3 – חוויית משתמש ושיפור רמת המשחק (13.5 - 2.6)**

* שיפור ממשק המשתמש להתאמת חוויית המשחק.
* פיתוח מנגנון משוב על הרמזים (לדוגמה: דירוג איכות לצורכי סטטיסטיקות).
* יצירת אפשרות לשחקן יחיד מול AI ושיפורים בהפעלת הבינה המלאכותית.

**שלב 4 – בדיקות ותיקוני באגים (3.6 - 23.6 )**

* ביצוע בדיקות תקינות של תהליכי המשחק והתקשורת ברשת.
* שיפור חוויית המשחק על פי משוב מהמשתמשים הראשונים.
* תיקוני ביצועים וסנכרון בזמן אמת באמצעות Firebase .

**שלב 5 – הכנת הפרויקט להצגה (24.6 - 15.7 )**

* בניית דמו עובד להצגת המשחק עם תרחישים שונים.
* כתיבת דוח מסכם ותיעוד הפרויקט.
* ביצוע הדגמות וליטושים אחרונים למערכת.

**תוצאות ראשוניות-**

* הוחלט על הארכיטקטורה המרכזית של המערכת.
* נבחרו טכנולוגיות לפיתוח: AI-OpenAI(GPT) , Backend–ASP.Net , Frontend-React SQL
* עוצבו מסכים ראשוניים (ללא פונקציונליות עדיין) .
* הוגדרו תפקידים במשחק ומנגנון המשחק תוכנן.

**אתגרים צפויים ואתגרים אלגוריתמיים –**

**יישום AI ליצירת רמזים אינטואיטיביים**

* בעיה: כיצד לגרום ל AI להבין קשרים בין מילים ולספק רמזים הגיוניים ומאתגרים.
* כיוון לפתרון: ניסוח שאילתות חכמות שיכוונו את ה – GPT לדוגמא לייצר רמזים איכותיים ומדויקים, הוספת מגבלות ולוגיקה מסננת כמו בדיקת אורך הרמזים או הימנעות ממילים מסוימות שעלולות לבלבל את השחקנים.

**סנכרון משחק מרובה משתתפים בזמן אמת**

* בעיה: כיצד לוודא שהמשחק פועל ללא השהיות ושכל השחקנים רואים אותו מהלך בזמן אמת.
* כיוון לפתרון: שימוש ב Firebase-Realtime-Database לניהול תורות ועדכון המסכים של כל המשתתפים בזמן רמת, בדיקת עומסים לוודא שהמשחק פועל בצורה חלקה עם מספר שחקנים במקביל.

**חווית משתמש – הצגת רמזים**

* בעיה: כיצד להבטיח שהשחקנים מקבלים רמזים ברורים ושהמשחק מספק משוב על איכות הרמזים.
* כיוון לפתרון: עיצוב ממשק משתמש ברור להצגת הרמזים בצורה נגישה ומהירה, יצירת משוב ויזואלי על איכות הרמזים(למשל חיווי צבעים) ולתת למשתמשים לדרג את הרמזים.

# תיאור מפורט של דרישות המערכת

## ייצוג מילולי של הדרישות הפונקציונאליות/הלא פונקציונאליות

בסעיף זה עליכם להציג את דרישות המערכת באמצעות טבלה 2 באופן הבא:

**דרישות פונקציונליות:**

* תפעוליות (operational)
* דרישות לממשקים חיצוניים: למכונות, למערכות אחרות (למשל מערכות מידע תפעוליות)

**דרישות לא פונקציונאליות:**

* + דרישות ביצועים של המערכת (למשל, דרישות לגבי זמן תגובה וביצוע פעולות).
  + דרישות אבטחת מידע (למשל, אילו אמצעים יינקטו כדי למנוע פריצה למערכת)
  + דרישות לממשק המשתמש
  + דרישות תשתית, טכנולוגיה וחומרה, בסיס הנתונים – DBMS, כלים הנדרשים לפיתוח ולתחזוקה

*טבלה 2. דרישות*

| **מס' הדרישה** | **סוג הדרישה** | **טיפוס** | **מודול/משתמש** | **נוסח הדרישה** | **הערות** |
| --- | --- | --- | --- | --- | --- |
| **1** | **פונקציונלית** | **O** | **מנהל משחק** | **המנהל יכול ליצור משחק חדש, להזמין**  **שחקנים ולשלוח הזמנות להצטרפות**  **למשחק.** | **חיוני להתחלת**  **המשחק.** |
| **2** | **פונקציונלית** | **O** | **שחקן** | **השחקן יכול להירשם לחשבון חדש**  **ולהתחבר למשחק על ידי הכנסת שם**  **משתמש וסיסמה.** | **תהליך התחברות**  **פשוט ונוח.** |
| **3** | **פונקציונלית** | **O** | **מנהל משחק,**  **שחקן** | **המנהל יכול להתחיל את המשחק,**  **ולעדכן את סטטוס המשחק (התחלה,**  **הפסקה, סיום) בזמן אמת.** | **דרישה קריטית**  **לניהול המשחק.** |
| **4** | **פונקציונלית** | **O** | **שחקן, מנחה AI** | **השחקן יוכל להגיש ניחושים ומילים בזמן**  **משחק לפי רמזים שהתקבלו מהמנחה.** | **כל שחקן חייב**  **לראות את שלביו.** |
| **5** | **פונקציונלית** | **O** | **מנחה AI** | **המנחה AI יוכל להפיק רמזים לשחקנים**  **ולהיות אחראי על ההתנהלות של**  **המשחק.** | **דרישה**  **לאינטראקציה**  **אוטומטית.** |
| **6** | **לא פונקציונלית** | **P** | **כל משתמש** | **המערכת תוכל לתמוך בשחקנים רבים**  **בו זמנית, עם זמן תגובה נמוך ואפס**  **השהיות בין שחקנים.** | **חיוני למניעת**  **פיגור**  **במשחק.** |
| **7** | **לא פונקציונלית** | **Q** | **כל משתמש** | **המערכת תדרוש הצפנת מידע אישי של**  **השחקנים, כולל פרטי ההתחברות.** | **עמידה בחוקי**  **פרטיות ואבטחת**  **מידע.** |
| **8** | **פונקציונלית** | **O** | **כל משתמש** | **המערכת תבצע שמירה אוטומטית על**  **נתוני המשחק, כך שבמקרה של ניתוק,**  **לא יאבד מידע על מצב המשחק.** | **חשוב לא להפסיד**  **נתונים במשחק.** |
| **9** | **פונקציונלית** | **O** | **כל משתמש** | **המערכת תציג את התוצאות לאחר סיום**  **המשחק, כולל נקודות הקבוצות ומי**  **ניצח.** | **דרישה חשוב**  **להציג את**  **המנצחים.** |
| **10** | **לא פונקציונלית** | **I** | **כל משתמש** | **הממשק יהיה נעים וקל לשימוש, עם**  **עיצוב אינטראקטיבי ושקוף לשחקנים.** | **חשוב להבטיח**  **חווית משתמש**  **איכותית.** |
| **11** | **פונקציונלית** | **O** | **שחקן** | **השחקנים יוכלו לתקשר אחד עם השני**  **במהלך המשחק דרך מערכת צ'אט, אך**  **לא עם קבוצת היריב.** | **חיוני לשיתוף**  **פעולה בתוך**  **הקבוצה.** |
| **12** | **לא פונקציונלית** | **M** | **כל משתמש** | **המערכת תוכל להתעדכן באופן**  **אוטומטי, לבצע תיקונים ושדרוגים,**  **ולבצע בדיקות איכות שוטפות.** | **יש לוודא תחזוקה**  **רציפה למערכת.** |
| **13** | **לא פונקציונלית** | **P** | **כל משתמש** | **המערכת תסנכרן את מצב המשחק בין**  **כל השחקנים בזמן אמת, כך שכל שחקן**  **יקבל את עדכון המילים והסטטוס בזמן**  **אמת.** | **דרישה קריטית**  **למניעת פיגור.** |
| **14** | **פונקציונלית** | **O** | **כל משתמש** | **המערכת תספק הודעות שגיאה ברורות**  **בעת תקלה, ותספק אפשרות למשתמש**  **להיכנס מחדש למשחק אם הוא נותק.** | **חשוב לטפל**  **בשגיאות בזמן**  **אמת.** |
| **15** | **לא פונקציונלית** | **H** | **כל משתמש** | **המשחק יתפקד ללא בעיות על דפדפנים**  **שונים, ויתמוך במכשירים שונים (PC,**  **טלפונים ניידים).** | **חיוני לא לפסול**  **מכשירים שונים.** |
| **16** | **פונקציונלית** | **D** | **כל משתמש** | **המערכת תשמור את פרטי המשתמשים,**  **כולל שם משתמש, סיסמה ומידע אישי**  **(כמו מייל) בצורה מוצפנת ובטוחה.** | **אבטחת מידע**  **קריטית.** |
| **17** | **פונקציונלית** | **D** | **כל משתמש** | **המערכת תעבור לשמור את כל נתוני**  **המשחק, כולל המילים שנבחרו, רמזים**  **שניתנו, ניחושים שהוגשו וסטטוס**  **המשחק בזמן אמת, כך שכל שינוי יישמר**  **באופן אוטומטי.** | **שמירה בזמן אמת.** |
| **18** | **פונקציונלית** | **D** | **כל משתמש** | **המערכת תאפשר לשחקנים לראות את**  **היסטוריית המשחקים הקודמים שלהם,**  **כולל תוצאות, דירוגים, ומילים ששוחקו.** | **שמירת נתונים**  **על היסטוריה.** |
| **19** | **פונקציונלית** | **D** | **כל משתמש** | **המערכת תוכל לשמור את המילים שזוהו על ידי כל קבוצה ואת כמות הזמן של כל סבב המשחק.** | **ניהול זמן במשחק.** |
| **20** | **לא פונקציונלית** | **D** | **כל משתמש** | **המערכת תאפשר גיבוי ושחזור נתונים של המשחקים במקרה של ניתוק או תקלה טכנית, כך שכל נתון יישמר באופן אוטומטי.** | **צורך בגיבוי**  **ושרידות נתונים.** |
| **21** | **פונקציונלית** | **Q** | **כל משתמש** | **המערכת תספק מערכת אבטחה**  **מתקדמת, כולל הצפנה לכל נתוני**  **המשתמשים ונתוני המשחק, ותוודא**  **שהמידע נשאר חסוי ופרטי.** | **עמידה בתקנים**  **בינלאומיים.** |
| **22** | **פונקציונאלית** | **I** | **כל משתמש** | **ניהול תורים בזמן אמת באמצעות fire base** | **לא מספיק טוב** |
| **23** | **פונקציונאלית** | **I** | **כל משתמש** | **המערכת תשתמש ב-REST API**  **לשיתוף מידע בין השרת לממשק הלקוח**  **בצורה של JSON.** |  |
| **24** | **פונקציונאלית** | **I** | **כל משתמש** | **המערכת תציג רמזים ותגובות מנחה**  **ChatGPT בצורה ברורה, כולל אפשרות**  **לשאול שאלות ולקבל תשובות.** |  |

## ייצוג דרישות פונקציונאליות באמצעות תרשימי use case

אם התרשים מכיל בועות רבות (למעלה מ 15), יש לפצל למספר תרשימים, לפי המשתמשים.

**ספציפיקציה של USE CASES**

יש לבחור (יחד עם המנחה) **לפחות 5 בועות משמעותיות**, ולספק להן את הספציפיקציה הנדרשת. בפרט, יש לפרט באמצעות טבלה לכל אחת מהבועות שנבחרו את הפרטים הבאים:

| **שם ה Use case** |  |
| --- | --- |
| **תיאור קצר** |  |
| **שחקנים** |  |
| **תנאים מקדימים** |  |
| **תנאים מאוחרים** |  |
| **טריגרים** |  |
| **תרחיש מוצלח** |  |
| **תרחיש אלטרנטיבי חלופי/כשלון** |  |

# תיאור מפורט של הפתרון/השיטה – תיכון (design)

פה יש לפרט את תיכון (design) של המערכת על פי שלושת השכבות הבאות: שכבת המידע, שכבת העיבוד והאלגוריתמיקה, ושכבת התצוגה.

## תיכון שכבת המידע

* הציגו את רשימת הטבלאות בהתאמה **או** את תרשים ה DSD

אם בחרתם להציג את רשימת הטבלאות, יש לפרט לכל טבלה: שם טבלה, תיאור קצר, פירוט השדות. לפרט מהם המפתחות הראשיים, מהם המפתחות הזרים ולאן מפנים.

* במידה וקיימים אילוצים מיוחדים על הנתונים בטבלאות, יש לציין אותם באופן מפורש.

## תיכון שכבת העיבוד והאלגוריתמיקה

* הציגו את ה DIAGRAM CLASS של המערכת.
* תארו את האלגוריתמיקה של המערכת.

אם האלגוריתם הנו תכנון של מכונת מצבים מורכבת, יש לתאר את מכונת המצבים באמצעות תרשים מצבים (state machine). אחרת, מומלץ להיעזר בדיאגרמת הרצף (sequence diagram) לתיאור האלגוריתם.

**אם נעשה שימוש באלגוריתם קיים (למשל, אלגוריתם שיבוץ):**

* + - יש להסביר מה מטרתו של האלגוריתם וכיצד הוא משתלב בפרויקט
    - יש להסביר מדוע נבחר אלגוריתם זה ולא אחר, ולתאר את אופן פעולתו
    - יש לתאר את אופן ההתאמה (אדפטציה) לצרכי הפרויקט – אם נדרשת התאמה
    - יש לתאר את אופן השימוש, מהו ה INPUT, מהו ה OUTPUT, מי הפרוצדורה שקוראת לאלגוריתם ומתי.

**אם נעשה שימוש באלגוריתם שפותח ע"י הסטודנטים לטובת הפרויקט:**

* + - יש להסביר מה מטרתו של האלגוריתם וכיצד הוא משתלב בפרויקט
    - יש להסביר מדוע נבחר לעשות פיתוח של אלגוריתם ולא להשתמש באלגוריתם קיים
    - יש לתאר את אופן השימוש, מהו ה INPUT, מהו ה OUTPUT, מי הפרוצדורה שקוראת לאלגוריתם.
    - יש לתאר אותו בפסאודו-קוד או בעברית מובנית

**אם נעשה שימוש באלגוריתם אחד או יותר ממשפחת ה data science/ ML:**

* + - יש להסביר מה מטרתו של האלגוריתם וכיצד הוא משתלב בפרויקט
    - יש להסביר מדוע נבחר אלגוריתם זה ולא אחר (למשל: רגרסיה לוגיסטית, KNN...), ולתאר את אופן פעולתו.
    - אם הוא עושה שימוש ב DATA יש לפרט את מקורות המידע ומהן פעולות העיבוד הנדרשות.

## תיכון שככת התצוגה: ממשק משתמש

* ציירו את מפת הניווט של המערכת

מפת הניווט מגדירה את שטף העבודה מנקודת המבט של המשתמש. מפת ניווט כוללת את ההיררכיה של מסכי המערכת ואת חיצי הניווט בין המסכים השונים. אם יש שוני מהותי בפונקציונאליות השימוש של פרופילים שונים של משתמשים – יש ליצר מפת ניווט נפרדת לכל פרופיל.

במידה ומפת הניווט גדולה מידי ואינה קריאה, יש לפצל את מפת הניווט לכמה עמודים (רצוי לפי משתמשים או פונקציונאליות)

* תארו באופן סכמטי את המסכים\* (layout) באמצעות כלי mockups.

לכל מסך יש לציין: לאיזו בועה בתרשים ה-UC הוא מתייחס, ואת אוסף הפעולות שמתבצעות במסך זה.

## אופן המימוש

פה יש לפרט על טכנולוגיות שעשיתם בהם שימוש, דוגמאות לקוד מרכזי, תכנית בדיקות ותוצאות של הבדיקות שביצעתם

* טכנולוגיות וכלים בשימוש
* דוגמאות לקוד מרכזי
* תכנית בדיקות ותוצאות

# סיכום ומסקנות

## השגת מטרות הפרויקט

פה יש לפרט עד כמה השגתם את מטרות הפרויקט כפי שהגדרתם, והתייחסות לחזקות וחולשות.

## תובנות שנלמדו

פה יש לפרט אילו תובנות, יכולות, כישורים רכשתם במהלך העבודה על הפרויקט

## הצעות לשיפור

פה יש לפרט הצעות לשיפור המערכת (אם קיימות)

## מבט לעתיד

פה יש לפרט רעיונות לעבודה עתידית והרחבת הפתרון לטובת תמיכה בפיצ'רים נוספים

# רשימה ביבליוגרפית

# נספחים

בחלק זה יופיעו כל הנספחים של המסמך: מסמכים שנאספו, שאלונים, צילומי מסכים של מערכות קיימות, טפסים, פרומפטים של שימוש ב AI וכיו"ב. בנוסף, אם קיים:

* קוד מקור (או URL לצפייה)
* מדריך משתמש
* רשימת מקורות (ספרות אקדמית, מדריכים טכנולוגיים, ומקורות נוספים ששימשו בעבודה).