# **טכנולוגיות אינטרנט מתקדמות - 61776 (WEB)**

**משימת בית מס' 1**

**להגשה עד 20.4.25 בשעה 23:59**

1. יש למנות מהנדס מערכת בכל צוות, אשר יהיה אחראי על הגדרת והקצאת המשימות בתרגיל זה.  
   נא לרשום את שם הסטודנט בתרגיל זה. על מהנדס המערכת לכתוב כיצד נעשתה חלוקת העבודה מול הצוות, מה היו המשימות של כל חבר צוות, האם היה ממשק בין חברי הצוות, והאם המשימות מולאו:

|  |  |  |
| --- | --- | --- |
| **שם חבר הצוות** | **משימות שהוקצו** | **משימות שהושלמו** |
| מהנדס: ראובן יצחקוב | 1. בחירת שם אתר וניסוח קצר  2. useCase  3. divergent thinking  4. convergent thinking  5. יצירת מסך התחברות | 1  2  3 |
| פלג כנפי | 1. ליצור פרסונה  2. לבצע ראיון  3. מפת אמפתיה  4. divergent thinking  5. convergent thinking  6. יצירת מסך ראשי | 1  2  3  4 |
| גל מויאל | 1. ליצור פרסונה  2. לבצע ראיון  3. מפת אמפתיה  4. divergent thinking  5. convergent thinking  6. יצירת מסך ראשי | 1  2  3  4 |
| עדן פורמן | 1. בחירת שם אתר וניסוח קצר  2. useCase  3. divergent thinking  4. convergent thinking  5. יצירת מסך התחברות | 1  2  3 |
| נופר בזק | 1. דרישות פונקציונליות  2. divergent thinking  3. convergent thinking  4. יצירת לוגו  5. בחירת עיצוב  6. בחירת צבעים | 1  2  4  5  6 |
| אביעד סלמה | 1. דרישות פונקציונליות  2. divergent thinking  3. convergent thinking  4. יצירת לוגו  5. בחירת עיצוב  6. בחירת צבעים | 1  2  4  5  6 |

קישור גיט של הצוות : <https://github.com/reuven-itzhakov/Advanced-Web-Technologies>

2. נושא הפרוייקט - Text to UML

3. בצעו תהליך של חשיבה עיצובית כפי שעשיתם בסדנה בהרצאה:

א. רשמו את שם האתר שנבחר, ופסקה קצרה של הסבר והקשר (קונטקסט).

Tx2UML

הסבר והקשר: השם שנבחר מרמז על יצירת UML Diagram על ידי טקסט.

ב. בצעו ראיון קצר עם דמות מרכזית (אמיתית) המייצגת משתמש במערכת. הגדירו את הפרסונה.ציירו empathy map.

**ראיון קצר עם הדמות המרכזית**

**מראיין:** היי דני, תוכל לספר לי איך אתה כותב דיאגרמות UML כיום?  
**דני:** בדרך כלל אני כותב את הקלאסים בטקסט ואז יוצר את הדיאגרמה בתוכנות כמו- visual paradigm אבל זה דורש הרבה עבודה.

**מראיין:** איזה קשיים אתה חווה בתהליך הזה?  
**דני:** זה לוקח הרבה זמן וזיכרון של איפה כל אלמנט נמצא בתוכנה. הייתי מעדיף להקליד טקסט פשוט כמו “Car is a class” ולקבל דיאגרמה מוכנה.

**מראיין:** מה אתה היית מצפה ממערכת שתהפוך טקסט לדיאגרמת קלאס?  
**דני:** שהמערכת תקבל טקסט ותבנה לי תרשים מסודר עם קשרים כמו ירושה, קומפוזיציה וכדומה.

**מראיין:** יש משהו שהיית רוצה לשלוט בו בתוך הדיאגרמה?  
**דני:** כן, למשל שמות הקלאסים, הסדר ביניהם, צבעים אולי בשביל להבין היררכיות או סוגי קשרים.

**הגדרת הפרסונה**

**שם:** דני כהן

**Job + Description:**

דני הוא סטודנט שנה ב' להנדסת תוכנה במכללת בראודה. במסגרת הלימודים, הוא נדרש ליצור דיאגרמות UML בפרויקטים שונים, בעיקר בקורסים של תכנות מונחה עצמים. הוא משתמש ב-Visual Paradigm לצורך יצירת הדיאגרמות, אך מרגיש שהתהליך גוזל ממנו זמן רב, במיוחד כאשר עליו לבצע עדכונים תכופים.

**Goals / Needs:**

יצירת דיאגרמות UML במהירות ללא צורך בשרטוט ידני.

המרה אוטומטית של טקסט UML לדיאגרמה ברורה.

אפשרות לערוך ולשמור את הדיאגרמה.

הורדה ושיתוף מהיר של הדיאגרמות לצורך הגשת עבודות ופרויקטים.

**Tasks:**

להזין טקסט המתאר מחלקות UML.

לקבל דיאגרמה מוכנה באופן אוטומטי.

לערוך ולשפר את הדיאגרמה במידת הצורך.

להוריד את הדיאגרמה ולהשתמש בה בפרויקטים.

**Scenario:**

דני עובד על פרויקט בקורס תכנות מונחה עצמים, בו הוא נדרש להגיש דיאגרמת מחלקות UML כחלק מהמסמך הטכני של הפרויקט.

במקום לבנות את הדיאגרמה ידנית ב-Visual Paradigm, הוא משתמש באתר Text2UML, שבו הוא פשוט מקליד טקסט המתאר את המחלקות והקשרים ביניהן ומקבל דיאגרמת UML מוכנה בלחיצת כפתור.

לאחר מכן, הוא מבצע את העריכות הנדרשות ולבסוף יוצר תמונה לצורך הגשת הפרויקט, וחוסך לעצמו זמן יקר בלמידה ובעבודה.

**Empathy map**

|  |  |
| --- | --- |
| **עושה**  משתמש בתוכנות כמו .visual paradigm  מחפש דרכים קלות ליצור דיאגרמה. | **אומר**  **"**זה לוקח לי הרבה זמן."  "אני רוצה לעשות זאת בקלות."  " לא אוהב לשרטט ידנית." |
| **מרגיש**  מתוסכל מיצירת דיאגרמות  מרגיש שזה יכול להיות יעיל יותר  מרוצה כשהכול מוצג בצורה ברורה | **חושב**  **"**אני מבזבז הרבה זמן על שרטוט."  "זה צריך להיות פשוט יותר."  "הבנה חזותית עוזרת לי להבין." |

ג. בצעו תהליך של divergent thinking. רשמו את כל הרעיונות שעלו.

1. המרה של טקסט טבעי לדיאגרמת קלאס (לדוגמה: "יש מחלקה בשם Student עם שם וגיל").

2. תמיכה בשפת קוד אמיתית (Java, Python וכו') שתתורגם לדיאגרמה.

3. אפשרות לשמור את הדיאגרמה כ-PNG, PDF אוVPP .

4. זיהוי אוטומטי של קשרים: ירושה, קומפוזיציה, Aggregation וכו'.

5. הצעות אוטומטיות להוספת קשרים חסרים בין מחלקות.

6. ממשק גרפי אינטראקטיבי שמראה את הקלאסים בזמן אמת.

7. תמיכה בשפות רבות – עברית, אנגלית, רוסית וסינית.

8. אופציה לשתף דיאגרמה עם חבר/מרצה/צוות.

9. עיצוב אוטומטי של הדיאגרמה לפי סגנון נבחר (Dark Mode / Light Mode).

10. שימוש ב-AI לתיקון טקסט לא מדויק (לדוגמה: "יש מחלקה של רכב עם מהירות").

11. אפשרות לעבוד בקבוצה על דיאגרמה בזמן אמת (כמו Google Docs).

12. היסטוריה של שינויים – Undo/Redo לכל פעולה.

13. התאמה אישית של צבעים, גופנים וסגנונות של הקלאסים.

ד. בצעו תהליך של convergent thinking. רשמו את כל השיפורים שעלו.

לאחר שלב של חשיבה מסתעפת, שבו גיבשנו מגוון רחב של רעיונות לפיתוח המערכת, עברנו לתהליך של חשיבה מתכנסת שמטרתו למקד את הפיתוח ברעיונות המרכזיים והישימים ביותר. בבחירת הרעיונות התמקדנו בקריטריונים של מעשיות טכנולוגית, השפעה על חוויית המשתמש, וכן התאמה להיקף ולמשאבים הזמינים לנו במסגרת הפרויקט.

בחרנו להתחיל ביישום של הפונקציונליות הבסיסית והחיונית ביותר:

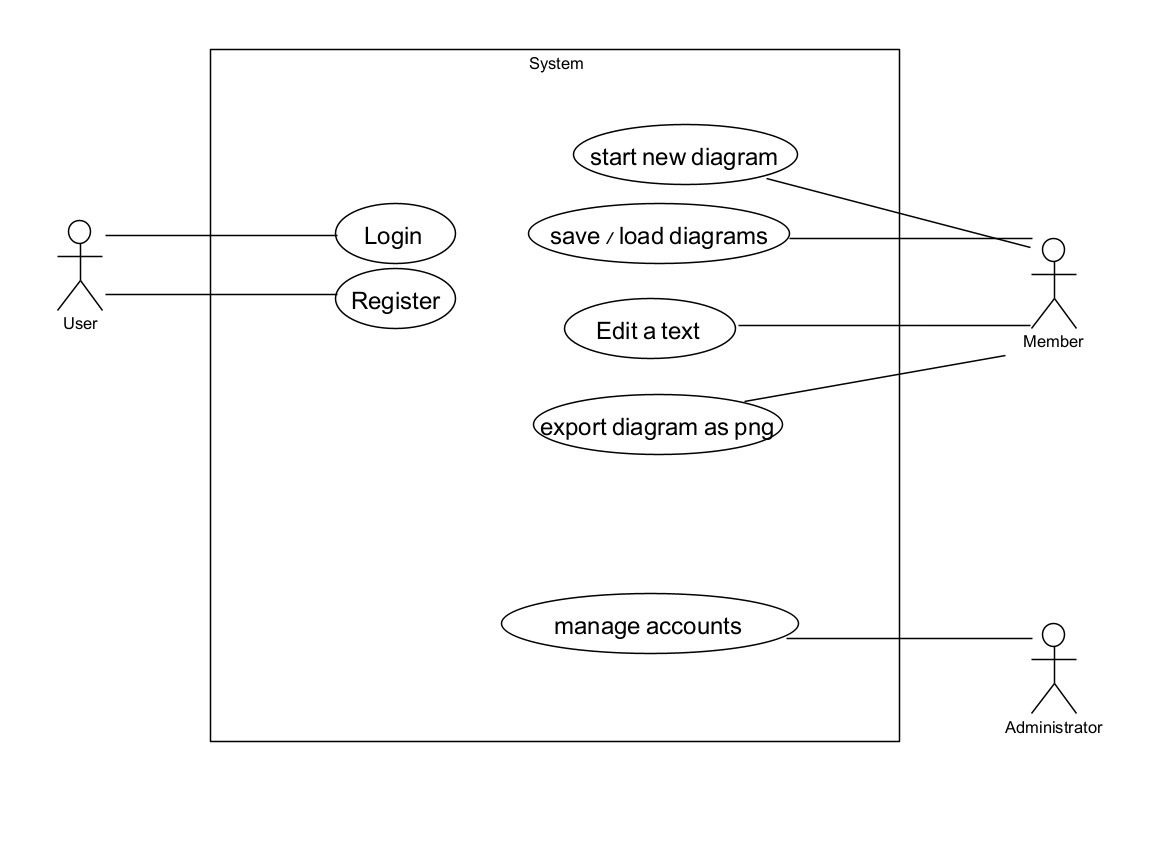
* המרה של טקסט טבעי לדיאגרמת קלאס (1) – זו ליבת הרעיון של המערכת שלנו, והיא מייחדת את הכלי לעומת תוכנות קיימות. המיקוד כאן הוא לאפשר למשתמשים להקליד טקסט פשוט ולקבל תוצאה ויזואלית מיידית.
* אפשרות לשמירת הדיאגרמה כקובץ PNG (3) – כדי שהמשתמש יוכל לייצא את העבודה שלו ולשתף אותה או להגיש אותה כקובץ סטטי.
* ממשק גרפי אינטראקטיבי (6) – כדי להציג את הדיאגרמה בצורה ויזואלית ברורה ונוחה לשימוש, כולל עדכונים בזמן אמת.
* תמיכה באנגלית בלבד (7) – מתוך שיקול של פשטות והפחתת מורכבות בפיתוח בשלב הראשוני. אנגלית היא השפה המרכזית בתכנות, ולכן נבחרה כברירת מחדל.
* עיצוב אוטומטי של הדיאגרמה לפי סגנון (9) – תורם לנראות ולחוויית המשתמש, במיוחד עבור משתמשים עם העדפות שונות (למשל מצב כהה או בהיר).
* היסטוריית שינויים (Undo/Redo) (12) – מאפשרת עבודה נוחה ונטולת חשש מטעויות, משפרת את חוויית המשתמש באופן משמעותי.
* התאמה אישית של צבעים, גופנים וסגנונות (13) – כדי לאפשר לכל משתמש להתאים את הדיאגרמה לצרכיו או להעדפותיו האישיות.

רעיונות אלו מהווים איזון בין ליבת הפונקציונליות לבין אלמנטים של נוחות ואסתטיקה, והם צפויים להביא לתוצאה שימושית, מרשימה, אך גם ברת ביצוע במסגרת הזמן והמשאבים שברשותנו. בהמשך, נוכל להרחיב את המערכת ולשלב פונקציות נוספות מרשימת הרעיונות המקורית.

ה. רשמו 5 דרישות פונקציונליות מרכזיות ו-5 דרישות לא פונקציונליות מרכזיות (באנגלית). יש לסווג את הדרישות הלא פונקציונליות לפי:  
<https://en.wikipedia.org/wiki/Non-functional_requirement>

|  |  |  |
| --- | --- | --- |
| # | **Requirement** | **Type** |
| 1 | The system shall be able to generate a class diagram | FR |
| 2 | The system shall allow adding inheritance between classes | FR |
| 3 | The system shall allow adding attributes to a class | FR |
| 4 | The system shall allow editing and deleting elements | FR |
| 5 | The system shall allow creating relationships between classes | FR |
| 6 | The user shall be responsible for adding a class | NFR |
| 7 | The user shall be responsible for the logical correctness of text | NFR |
| 8 | The interface shall contain tooltips to guide the user | NFR |
| 9 | The system shall support large-scale diagrams | NFR |
| 10 | The system shall switch to dark mode after 19:00 | NFR |

ו. הציגו תרשים USE CASE של האתר.



3. יש להציג LAYOUT - דף HTML הכולל מסכים ראשוניים לפרויקט שלכם. הדף יכלול שימוש ב pure JS, וכן שימוש ב Tailwind.מומלץ להתבסס על התרגולים.לצורך הצגת המידע יש להשתמש ב Fake data, להמחשת העבודה עם הנתונים .

יש להגיש את כל התוצרים בתיקיית גיט של הצוות.