

שלב 1 – לוגיקה עסקית

1. מוצר הקוד הראשון שנממש יהיה 'מחשבון פיזיקה' של הזבובון.
2. לצורך פיתוח מהיר ובצעדים קטנים, נניח כי הזבובון מתנהג כגוף בתעופה בליסטית קלאסית כפי שנלמדת בפיזיקה בתיכון. א. **להרחבה על תנועה בליסטית** ב. נתעלם מהשפעות הרוח בשלב הראשון.
3. עליכם לממש מוצר תוכנה אשר מבצע את חישובי הפיזיקה המתאימים.

המוצר מצורף. העליתי את המימוש שביצעתי למוצר התוכנה ב-Git.

קלט:

- מהירות התחלתית בעת הירי של הזבובון
- זווית התחלתית אל מול האופק
- גובה התחלתי מעל פני הקרקע

פלט:

- מיקום הזבובון בעת פגיעתו בקרקע
- מהירות הפגיעה בקרקע
- **בונוס:** זווית הפגיעה בקרקע

4. **שאלה:** איך נוכל לוודא כי המוצר שבנינו מוסר ללקוח תשובות נכונות? אילו מקרי קצה עלולים לצוץ בעת שימוש במערכת? כיצד נתמודד איתם?

תשובה: נוכל לוודא שהמוצר שבנינו מוסר ללקוח תשובות נכונות על ידי כתיבת מחלקה של טסטר למוצר. מקרי קצה שיכולים לצוץ בעת השימוש במערכת הם:

1. זווית שהיא 0 או 360.
2. גובה 0.
3. מהירות 0.

כל משתנים מסוג double.

נתמודד עם כל אלה לפי המחשבון שכתבנו כי זה אכן יחשב את התוצאות הדרושות אך אם ינסו להכניס ערכים שליליים בגובה או במהירות, או זווית שאינה בין 0-360 אז צריך להוציא הודעת שגיאה מתאימה.

5. **שאלת בונוס:** המודל הפיזיקלי שאנחנו מממשים כאן לא מתאר בצורה מדויקת לחלוטין התנהגות מציאותית. כיצד נשפר את המודל כך שיהיה מדויק יותר? אילו תנאי סביבה של המציאות ניתן יהיה לדעתכם לממש במודל?

תשובה: אפשרי לשפר את המודל בכך שנתייחס בתנועה הבליסטית לחיכוך עם האוויר דבר שהתעלמנו ממנו קודם לכן.

שלב 2 – ארכיטקטורה

1. עתה נרצה להנגיש את הלוגיקה העסקית שפיתחנו על גבי הרשת המבצעית.

2. נניח כי הרשת המבצעית והאינטרנט האזרחי זהים לחלוטין מבחינת משאבים ותשתיות.

3. בהתאם לסטנדרט השולט כיום בעולם, אנחנו מעוניינים לייצר מערכת **מבוזרת**, כלומר, מבוססת Micro-Services על בסיס k8s.

4. **שאלה:** כיצד תבחרו לממש את הארכיטקטורה? ניתן לצרף בשלב זה גרף או שרטוט המתאר את Flow המידע במערכת.

תשובה: Flow המידע במערכת- צורף כתמונה.

5. עליכם לממש מוצר תוכנה נוסף, **רשתי**, אשר מאפשר גישה למחשבון הפיזיקלי שמומש בסעיף הקודם.

a. המוצר החדש יוכל לקבל בקשת HTTP.

b. המוצר החדש יחזיר תגובת HTTP אשר תכיל את התשובה לחישוב.

העליתי את מימוש התוכנה שביצעתי ל-Git. מימשתי בקשות rest שעובדות מעל ה-http.

6. **שאלה:** מה על המערכת לעשות אם נשלחה בקשת חישוב לא צפויה?

תשובה: אם בקשת החישוב אינה ניתנת לחישוב אז המערכת צריכה לשלוח הודעת שגיאה בהתאם. לדוגמא- אם אחד הערכים המתקבלים הוא שלילי צריכה להיות הודעת שגיאה כי המהירות הגובה והזווית לא יכולים להיות שליליים.

7. **שאלת בונוס:** בהתחשב בסיווג המערכת, כיצד נוכל להגן על הנתונים שזורמים במהלך התקשורת?

תשובה: אפשר להשתמש בבקשות https. בקשות שהן יותר מאובטחות מאשר בקשות ה-http. דבר שיגן על הנתונים הזורמים במהלך התקשורת בצורה טובה יותר.

8. **שאלת בונוס 2:** ישנו סיכוי כי 2 או יותר משתמשים יבקשו את אותו החישוב (אותו

קלט). כיצד ניתן לייעל את המערכת לאור זאת?

תשובה: נוכל לייצר cash שהוא בעצם מאין data base שכל פעם שמישהו יבקש חישוב מסוים החישוב יכנס ל-cash ולכן כאשר מישהו אחר יבקש את אותו החישוב הוא יחזיר את החישוב שבוצע קודם לכן, ללא לחשב זאת שוב. דבר שייעל את המערכת.

9. **שאלת בונוס 3:** אנו צופים כי עלולים להיות פרקי זמן בהם השימוש במערכת יהיה אינטנסיבי (המון משתמשים בבת אחת). כיצד נוכל לשפר את המערכת כך שתוכל לתת מענה מהיר למשתמשים רבים?

תשובה: אפשר להוסיף עוד מופע של server ב-k8s ולשים אובייקט שנקרא load balancer שהוא בעצם מקבל בקשות מה-client ומעביר ל-server הכי פחות עמוס. ובכך מנהל את קבלת הבקשות מה-client באופן יותר יעיל וגם את ביצוען.

שלב 3 – אפליקציה

1. לאחר הטמעת ובדיקת המערכת, נרצה להציגה לחייל הקרבי המפעיל את הזבובון באמצעות ממשק משתמש נוח ופשוט.

2. על בסיס הארכיטקטורה שבחרנו, נרצה להקים Single Page Application. האפליקציה תציג יכולת להזין קלט למחשבון הפיזיקלי ובעקבותיו תציג פלט.

3. שרטטו או ממשו ממשק משתמש אשר יאפשר שימוש ויזואלי במערכת. הממשק נדרש להציג את היכולת להכניס את הקלט ולהציג את הפלט של המחשבון הפיזיקלי.

שרטוט ממשק המשתמש- צורף כתמונה.

4. **שאלה:** על מה נשים דגש בעת פיתוח ממשק המשתמש?

תשובה: בעת פיתוח ממשק המשתמש נשים דגש על כך שיהיה ברור(קל) לתיפעולם של החיילים ושתינתן הודעת שגיאה אם יש צורך בכך.

5. **שאלת בונוס:** איך נפתח את הממשק? באילו טכנולוגיות? מאילו אבני בניין נבנה את

ממשק המשתמש?

תשובה: נפתח את הממשק בטכנולוגיית react. אבני הבניין שמהם נבנה את ממשק המשתמש יהיו- כפתורים, בלוקים להזנת נתונים וטיפול אירועים בחלקה נפרדת שאז יתבצע בעצם החישוב.

6. **בונוס 2:** ציירו את מעוף הזבובון באפליקציה.

תשובה: מפני שעשיתי עד עכשיו במטלה הזאת הכל בסביבת פיתוח של java אעשה גם את האנימציה של מעוף הזבובון בסביבה זאת על אף שהיא לא הכי נוח לעשות בה אנימציות.

את האנימציה הזאת ב-java הייתי עושה עם applet.

Applet היא מחלקה ב-java שאפשר לממש אותה בכדי להזיז תמונה באופן מסוים במקרה הזה את הזבובון במסלול המחושב על ידי המחשבון שיצרתי בשלב 1.

****בשל חוסר הזמן שלי בגלל שאני טס ביום שלישי(26.7) לכנס מדענים צעירים בלונדון ואני הולך לדבר שם על אסטרופיזיקה שבתחום זה עשיתי עבודה בטכניון. אז נורא נהנתי מהצד הפיזיקלי במשימה וגם מהצד הטכני של המחשבים אך לא הצלחתי לממש את הבונוס השני בשלב 3(האפליקציה), אך בכל זאת רציתי להראות שאני כן יודע את התאוריה מאחורי זה.**