**המחלקה להנדסת תוכנה**

**דו"ח הצעה לפרויקט סופי**

**קורס מדעי המחשב חלק ב'**

**תשפ"ה**

**תוכנית למיון מקרי חירום בבית החולים**

**מאת** תולין סלהב 329388003

**תאריך ההגשה**

# 14.5.2025

**מרצה הקורס**

**ספאא נתשה**

https://github.com/ **קישור למאגר קוד**

## תאור כללי של הפרויקט

פרויקט זה עוסק בפיתוח מערכת מיון חירום לבית חולים, שמטרתה לסייע בזיהוי וסיווג מקרי חירום רפואיים על פי תסמיני המטופל.

המערכת מאפשרת רישום פרטי מטופלים, קביעה אוטומטית של רמת הדחיפות, חישוב משך שהיית המטופל בבית החולים, ושמירת היסטוריית הנתונים לקובץ חיצוני לצורכי מעקב עתידי.

הפרויקט מדגים יישום של עקרונות תכנות מונחה עצמים (OOP) בסביבת פיתוח מקצועית, תוך דגש על סדר, ניהול נתונים וכתיבה נכונה של קוד.

## מטרות הפרויקט

.יישום ושימוש במבני נתונים כגון ArrayList לצורך אחסון וניהול מידע רפואי

.בנייה ותחזוקה של מחלקות (Classes) ואובייקטים בהתאם לעקרונות ה-OOP

.תכנון המערכת תוך הסתמכות על תרשימי UML מפורטים ומסודרים

.עבודה עם סביבת הפיתוח IntelliJ IDEA ושימוש בכלים לניהול גרסאות כמו GitHub

הוספת פונקציונליות מתקדמת כגון חישוב סטטיסטיקות על מקרי החירום ותיעוד זמני כניסה ויציאה של מטופלים.

## פונקציונליות מתוכננת

הוספת מטופלים חדשים למערכת על ידי הזנת מספר זהות, מין ותסמין עיקרי.1

קביעת עדיפות המקרה באופן אוטומטי על סמך התסמין שנבחר.2

חישוב ורישום זמן כניסה ויציאה של המטופל .3

שמירת היסטוריית מקרי החירום לקובץ חיצוני לצורכי תיעוד ומעקב.4

ספירת מספר המקרים בכל רמת עדיפות (גבוהה, בינונית, נמוכה) והצגת התוצאות.5

מתן המלצה למטפל בהתאם למצב (לדוגמה: “פנה מיידית לחדר מיון” או “יש לעקוב תוך 24 שעות”).6

הצגת רשימת כל המטופלים עם פרטיהם ומצבם הרפואי .7

**מבני נתונים מתוכנני ם**

שימוש ב-ArrayList לשמירת רשימות של מטופלים

שמירת רשימה של מקרים לפי עדיפות (גבוהה, בינונית, נמוכה)

יצירת היסטוריית מקרי טיפול (History) בקובץ חיצוני

ספירת מספר המקרים בכל רמת עדיפות

## טכנולוגיות ושפת תכנו ת

שפת תכנות: Java

סביבת פיתוח: IntelliJ IDEA

ניהול גרסאות: GitHub

## חלוקת עבוד ה בין חברי הקבוצה

אני עושה את הפרויקט לבד

## לוח זמנים משוער

|  |  |
| --- | --- |
| **תאריך** | **משימה** |
| 18.5.2025 | לימודGitHub והיכרות עם כלי ניהול גרסאות |
| 21.5.2025 | תכנון מבנה המחלקות ותכנית בדיקות בסיסית |
| 1.6.2025 | כתיבת מחלקתExpense והוספת תמיכה בסיסית בנתונים |
| 8.6.2025 | בניית תשתית ניהול ההוצאות ללא GUI |
| 15.6.2025 | תכנון להתחיל לכתוב את הקוד |
| 22.6.2025 | שילוב גרפים וממשק עם המערכת |
| 29.6.2025 | בדיקות סופיות, תיקונים והגשת הפרויקט |

**אתגרים צפויי ם**

## ניהול קשרי רבים-לרבים בין סטודנטים לקורסים ובין מורים לקורסים.

## טיפול נכון באימות קלט מהמשתמש (ולידציה).

## שמירה על תקינות המידע בזמן עבודה עם קבצים.

## עבודה שיתופית מסודרת באמצעות GitHub. נספחי ם

קוד ההתחלה למחלקת Expense