



תאריך ההגשה 31.8.2021

מערכת מידע למסעדת JavaEat:

תיאור כללי:

מסעדת JavaEat היא מסעדה חדשה שנפתחה בחיפה בתקופת הקורונה, ובהתאם לכך המודל העסקי שלה מתבסס על משלוחים. מנהל המסעדה החליט לפתח מערכת מידע כדי לייעל ולהתאים את תהליכי העבודה ולשפר את רווחיות העסק, מנהל המסעדה בחר בכך להיות מפתחי מערכת המידע של המסעדה. המסעדה צריכה להתמודד עם מאות משלוחים והזמנות רבות ביום ללקוחות רבים, כל הזמנה כוללת מנות שמכילות מרכיבים, והצוות כולל טבחים ושליחים. המסעדה מחלקת משלוחים לפי איזורים. מערכת המידע תצטרך לעזור לעובדי המסעדה לארגן משלוחים, ולחשב רווחים כלכליים. כעת, המסעדה מבקשת מכם לממש ממשק גרפי ללקוח ובנוסף לשמור את נתוני המערכת ולדעת לקלוט אותם.

מטרות

- עליכם לבנות מסכים/טפסים אשר יתמכו בפונקציונליות הבאה:
 - קבלת נתונים ליצירת אובייקטים במערכת. כאן יש להשתמש בכל מתודות ההוספה add למיניהן, בנוסף להצגת הודעות הצלחה וכישלון אינפורמטיביות.
 - קבלת נתונים למחיקת אובייקטים במערכת בהתאם לכל מתודות removed בנוסף להצגת הודעות הצלחה וכישלון אינפורמטיביות.
 - הפעלת כל השאילתות במערכת, והצגת התוצאות (הפליטים) של המערכת ושל השאילתות באמצעות מסכים המאפשרים הצגה נוחה למשתמש.* כל הפונקציות של המחלקה Restaurant צריכות להיות ממומשות כממשק למשתמש (באמצעות מסכים) שיהיה נוח לשימוש.
- עליכם לבנות פונקציונליות של שמירת נתוני המערכת ושיחזורם בהפעלה הבאה מתוך קובץ "Rest.ser", שבעצם ישמור את מצב האובייקט Restaurant על ידי ממשק Serializable. קובץ זה ישמש כסוג של Database.
- עליכם לטפל בחריגות כך שתוצג הודעת שגיאה (פופ-אפ) למשתמש להצגת הודעות שגיאה מתאימות.
- עליכם לבנות מחלקות טיפול בחריגות אפשריות (משלכם), לצורך זיהוי והצגת שגיאות. את המחלקות האלה יש למקם בחבילה שקיימת Exceptions
- עליכם לבחור את הרכיב המתאים ביותר עבור כל פריט במסכים שלכם למשל שדות טקסט, רשימות בחירה וכו...



תרגיל בית 3



לדוגמה:

- 5.1. אם רוצים לאפשר למשתמש לבחור ערך חוקי מתוך רשימה סגורה עליכם לתת לו לבחור מתוך הערכים הללו ולא לתת לו להקליד באופן ידני.
- 5.2. אם רוצים לאפשר למשתמש להקליד טקסט מבלי שיוצג על המסך (לדוגמא סיסמה)

תהליך העבודה

1. כאשר המערכת מופעלת היא בודקת אם קיים קובץ אחסון serializable בשם Rest.ser. אם הקובץ קיים אז המערכת תטען ממנו את אובייקט בית החולים עם כל הנתונים.
2. לאחר הבדיקה יפתח מסך ההתחברות למערכת בו נעשה כניסה login, באמצעות שם משתמש וסיסמה, הנתונים ייבדקו מול המערכת שתזהה את המשתמש ואת סוגו ותאפשר את האופציות הרלוונטיות לפי סוג המשתמש (הרחבה על סוגי משתמש בהמשך).
3. לאחר התחברות למערכת יפתח המסך הראשי שיקלוט:
 - 3.1. ניווט למסכים (חלונות) אחרים באמצעות תפריטים נוחים. למשל: חלון הפעלת שאילתות, חלון הזנת נתונים, חלון הצגת האובייקטים במערכת וכו...
 - 3.2. על כל המסכים להיות מסכים פנימיים של המסך הראשי
 - 3.3. כל ההודעות הקופצות הן פנימיות שייכות לחלון המערכת.
 - 3.4. שמירה של מחלקת בית החולים Restaurant לקובץ אחסון בשם Rest.ser.
 - 3.5. יציאה מהמערכת.

סוגי משתמשים

1. מנהל המסעדה (Manager):

משתמש זה יכול לבצע את כל האופציות במערכת כולל כל השאילתות והצגת כל הנתונים.

- 1.1. הפעלת כלל מתודות ההוספה במערכת.
 - 1.2. הפעלת כלל מתודות ההסרה במערכת.
 - 1.3. עדכון כלל פרטי המערכת.
 - 1.4. הפעלת כלל השאילתות במערכת.
 - 1.5. צפייה בכלל נתוני המערכת. (אם נדרש פרמטר, יוכנס כקלט)
- המשתמש מזוהה ע"י שם משתמש וסיסמה הקבועים בקוד והם manager.



2. לקוח מסעדה (Customer):

לקוח מזוהה על ידי שם משתמש וסיסמה שמוגדרים לו אישית.

הלקוח יכול להפעיל במערכת את הפעולות הבאות:

2.1. הצגת כלל ההזמנות של הלקוח וניהולן כולל מחיקה/הוספה. (כולל מנות והמרכיבים בהן)

2.2. ניהול סל קניות (shopping cart).

2.3. הוספת/הסרת מנה להזמנה. (של הלקוח)

2.4. יכולת לשנות רכיבים במנה. (ע"י יצירת אובייקט מנה חדש)

2.5. עדכון פרטים אישיים של הלקוח.

2.6. שאילתות שיוכל להפעיל : `getRelevantDishList` (בהתאם ללקוח עצמו), `getCookByExpert`, `getPopularComponent`.

2.7. הצגת תפריט המסעדה. (כלל המנות והמרכיבים בהן)

שימו לב -חלק מהרשאות ההתחברות כוללות מסכים זהים - נסו לעשות בהם שימוש חוזר ולהימנע משכפול קוד.

בנוסף, שימו לב מה כל משתמש רשאי לעשות ועל אילו אובייקטים אחרים הוא רשאי לבצע פעולות.

עליכם לבנות את המערכת כך שניתן יהיה להפעילה ע"י הקשה כפולה משולחן העבודה בעזרת jar file אותו ניתן ליצור דרך ה Eclipse. (ניתן לשים בתיקייה עבודה משום שנדרשת עבודה עם קבצים)



תרגיל בית 3



חריגות

עליכם לבנות מחלקה לכל סוג חריגה שעלולה להתרחש במערכת שתטפל בחריגה זו- למשל הוספת טבח קיים או קלט לא תקין.
טיפול בסוגים שונים של חריגות:

1. חריגות לוגיות שקשורות למערכת שלכם ספציפית אך אין חריגה מתאימה לכך Java.
2. עליכם להחליט עבור כל מתודה במחלקת Restaurant את סוגי החריגות הלוגיות שהיא יכולה לזרוק, ייתכן שכבר יש לכם ממוש מתרגיל קודם. עליכם להוסיף במידת הצורך.
3. עליכם לבנות את החריגות בהיררכיית הורשה מתאימה של מחלקות היודעות לטפל בסוגים שונים של חריגות לוגיות.

בונוס (30 נק') :

- חלק א' (עד 20 נק')
- ניתן לקבל עד 20 נקודות בחלק הזה לפי הפירוט להלן, לכל חלק יש ניקוד מקסימלי ולא ניתן לקבל יותר מ-20 נקודות:
1. אפשרות הוספת תמונה לכל Customer על ידי משתמש מורשה, בכניסה למערכת תופיע התמונה שלו באופן קבוע בפינה השמאלית העליונה של המערכת. גם כן בכל רשימה שתציג Customer יהיה ניתן לצפות/להיכנס לתמונה של המשתמש. (10 נק')
 2. אפשרות כתיבה למסמך (word ושיראה רשמי) של רווחים ממנות לפי מתודת `getProfitRelation`. (10 נק')
 3. מימוש כל החלונות בתרגיל כך שיתאימו את עצמם לרזולוציית המסך של המשתמש (ממשק משתמש רספונסיבי) (10 נק').
 4. במסך הצגת הנתונים, תינתן אפשרות לסנן את התוצאות לפי כל אחד מהפרמטרים של האובייקט. (לדוגמא: אפשרות לסנן מנה לפי מחיר או רגישות) **נדרש לממש סינון לפי כלל הפרמטרים.** (10 נק').
 5. במסך הראשי **הרקע** יהיה סרטון מהמסעדה (שרץ בלולאה). (10 נק').
 - שימו לב: לא להכניס סרטון שניתן להפעיל, אלא הרקע יהיה הסרטון.
 6. ממשק אינדיקטיבי – יתריע למשתמש בזמן הזנת הקלט האם הוא תקין או לא. לדוגמא: כאשר המשתמש ממלא שדה של סיסמא הוא ישר מקבל אינדיקציה של: סיסמא חלשה/בינונית/חזקה כולל אייקון מתאים שמתאר את זה. (ירוק – טוב, אדום – לא טוב וכו') (10 נק').
- (רק לממש את הדוגמא זה לא מספיק, **הממשק כולו צריך להיות אינדיקטיבי**)



חלק ב' (עד 10 נק')
חלק זה הינו חופשי, כלומר ניתן לקבל עד 10 נקודות על מימוש מעבר למבוקש, כגון: יצירתיות במימוש, תכנות אלגנטי, יעיל ועיצוב מושקע במיוחד (לא ברירת מחדל) ניתן להוסיף שימוש באנימציה צלילים – , sound effects תמונות רקע, חלונות זזים, מסך קביעת הגדרות או בקיצור תהיו יצירתיים.

**** עליכם להגיש קובץ PDF ביחד עם הפרויקט שיכיל את כל התוספות שהחלטתם לממש ולא מופיעות בדרישות**

הערות חשובות

- יש להקפיד על תיעוד והסבר של פעולותיכם בקוד ולשמור על המוסכמות של Java.
 - הגשה של פרויקט **שלא נוכל** להריץ תגורר הורדה משמעותית של נקודות מהציון. באחריותכם לבדוק ולוודא שהפרויקט רץ על כמה מחשבים שונים.
 - בתרגיל זה אין קליטת נתונים מקובץ קלט, כל הנתונים צריכים להיקלט דרך המסכים.
 - אובייקט של מחלקת Restaurant נוצר פעם אחת במערכת, כאשר עולה המסך הראשי. לאחר מכן, המערכת תקרא ותעבוד עם קובץ זה קבוע.
 - אסור** ליצור מופעים נוספים של מחלקה זו, לכן יש לדאוג שהאובייקט יעבור. הפתרון הנפוץ ביותר לכך הוא שימוש ב Singleton.
 - עליכם לממש את כל מחלקות ה GUI בתוך package שייקרא View.
 - עליכם למקם את כל המחלקות המטפלות בחריגות לוגיות ב package - שייקרא Exceptions.
 - יש לדאוג לכך שניתן יהיה לפתוח את קבצי הקוד שלכם דרך Eclipse.
 - יש לשים דגש על ממשק משתמש נוח נעים לעין, נקי למשתמש המהווה חלק גדול מהציון.
 - את הפרויקט ניתן לממש ב Eclipse, וניתן להשתמש בספריות ה GUI של java FX או swing.
- מומלץ מאוד להשתמש ב java FX.**
- על התרגיל להתבסס על פתרון תרגיל 2 גדול או לחילופין על בסיס הפתרון שלכם בהנחה ותיקנתם כל טעות שנמצאה וכמובן הוספת שאילתות הבונוס למי שלא מימש.
 - ניתן להגיש את התרגיל ביחיד/בזוגות/בשלוש.
- הציון הסופי יחושב לפי המקדם הבא:
- א. יחיד – ציון*1.1
 - ב. זוג – ציון*1.0
 - ג. שלשה – ציון*0.9



תרגיל בית 3



זאת אומרת: בהינתן יחיד יקבל ציון 100 זה ישוקלל ל110, זוג שיקבל 100 זה ישוקלל ל100, ושלושה שתקבל ציון 100 זה ישוקלל ל90.

- ✓ יש להגיש את התרגילים באתר ה-moodle בתיבת ההגשה בפורמט zip :
- JavaProject_Ex3_ID1_ID2.zip (כך שיכיל גם את קובץ ההרצה jar). כאשר ID1 ו-ID2 הם תעודות הזהות של שני הסטודנטים המגישים (במידה וההגשה היא בזוגות יש להגיש פעם אחת מהמודל של אחד מהסטודנטים). שימו לב, הגשה בפורמט אחר עלולה להוביל לחוסר בדיקה של התרגיל
- ✓ אם הוספתם פתרון לחלק של הבנוס נא לצרף קובץ PDF\WORD עם הסבר על ההוספות שנעשו, (הקובץ יהיה בתוך קובץ ההגשה JavaProject_Ex3_ID1_ID2.zip).
- ✓ שימו לב כי יש להגיש קובץ מכוון של הפרויקט ע"י פקודת Export מתוך eclipse ולא בדרך אחרת
- ✓ שימו לב - תרגיל שלא ירוץ או לא יתקמפל יגרור ציון.
- ✓ לכל חלק יפתח פורום, ויש לשאול שאלות לגבי החלקים רק דרך פורום החלק המיועד (ולא במייל)

בהצלחה!