דבר אלי UML ית

מחבר: איציק סיון

רקע

בעלי מקצוע שונים נבדלים ביניהם, בין השאר על ידי השפה שמאפשרת להם לתקשר ולהתבטא בצורה המקצועית המדויקת והעקבית ביותר. לולא שפה זו חלקם היה נשאר בתקופת האבן.

דוגמא א: מוסיקאים(מלחינים, נגנים, זמרים, מנצחים, מעבדים...) מתקשרים ביניהם באמצעות תווים. תארו לעצמכם כיצד היה מתפתח עולם המוסיקה ללא תווים? מה היה עושה מוצרט ללא תווים? זה לא אומר שאין עדיין מספר מוסיקאים שעדיין לא יודעים לקרוא/לכתוב תווים ונחשבים למוסיקאים מעולים

דוגמא ב: הנדסת בניין(ארכיטקטים, מהנדסי בניין, קבלנים, מעצבי מטבחים, מהנדסי חשמל...) מתקשרים ביניהם באמצעות שרטוטים המופקים בסיוע תוכנות CAD למיניהם(Computer Aided Design) כגון Autocad או RealCad . כיצד לדעתכם היו בונים את מגדלי עזריאלי בלי שרטוטים?

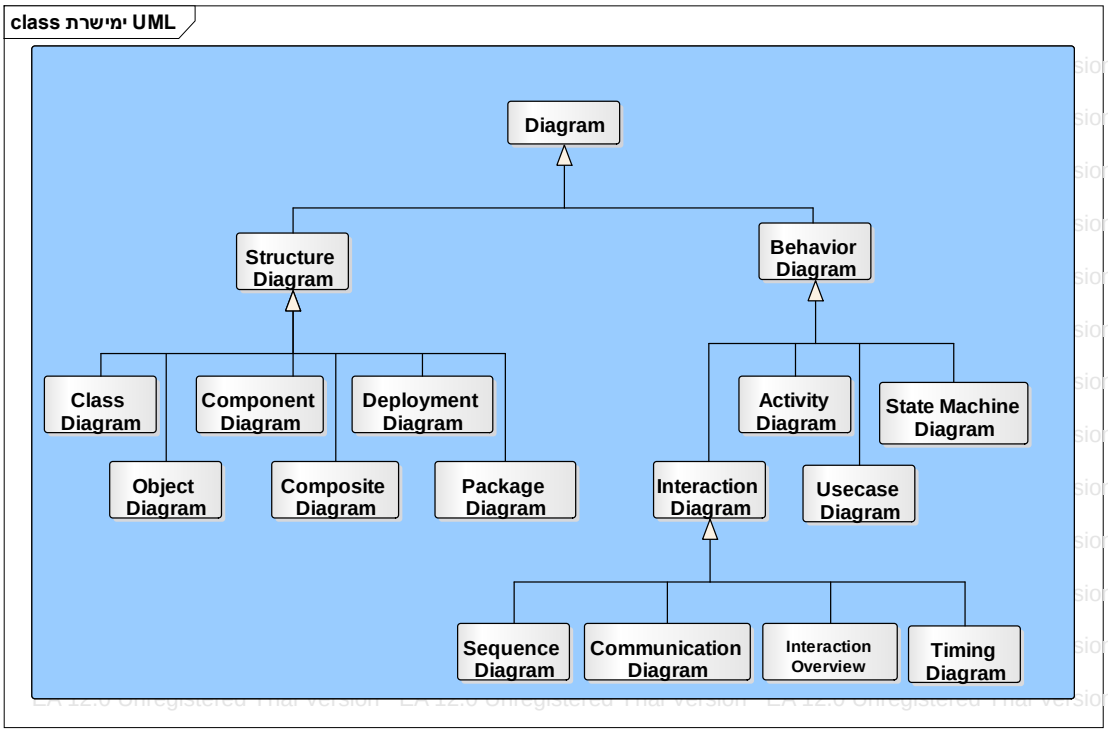
דוגמא ג: כימאים מתקשרים ביניהם באמצעות נוסחאות
כיצד היו מרכיבים פצצות אטום בלי נוסחאות?(אל תגלו לאירנים)
כיצד היו מסתדרים בחברת טבע ללא נוסחאות

ומה קורה אצלנו אנשי ה IT? למרות העובדה שאנחנו שייכים לענף שנקרא הנדסת תוכנה והנדסה משמעותה דיוק ומקצוענות הרי שהכלי העיקרי שלנו מזה עשרות שנים הינו מעבד תמלילים כגון Word.
Word למיטב ידיעתי נועד למזכירות, עיתונאים, סופרים, עורכי דין, היסטוריונים...
Word עם כל הכבוד(ויש כבוד) אינו מתיימר ולא יכול להיות כלי למהנדסים (נסו לתכנן בניין עם Word)

UML

Unified Modeling Language

שפת UML הוגדרה לראשונה ב1994 על ידי יזמים פרטיים
בשנת 1998 הוקמה ועדה אמריקאית-OMG (Object Management Group) שמטרתה הינה להגדיר שפה תקנית בינלאומית לעולם הנדסת התוכנה. הוועדה בחרה ב UML
שפת UMLמאפשרת ל"שרטט" את המערכת לפני שמעבירים אותה לקבלנים(סליחה-מפתחים)
השפה מבוססת על 13 תרשימים(תרשים אחד שווה אלף מילים)
התרשימים מסייעים בהפקת מסמכי אפיון, עיצוב וארכיטקטורה של מערכות תוכנה.
חשוב לציין , UML אינה מתודולוגיה לפיתוח מערכות!
UML משמשת כשפה משותפת ל : מנהלי פרויקטים,מנתחי מערכות, מעצבים, מפתחים, אנשי בדיקות
להלן מפה היררכית של התרשימים שמרכיבים את השפה



למרות שיש 13 תרשימים מנתח המערכות משתמש ב80 אחוז מהזמן ב 4 תרשימים.

בעקבות הגדרת שפת UML כתקן בינלאומי קמו חברות תוכנה רבות שמימשו את השפה בכלים יעודיים
כיום קיימות בשוק לפחות 30 חברות שמספקות כלי UML
החברות המובילות על פי חברת המחקר גרטנר הינן:

1. IBM עם מוצר בשם Rational Rose
2. SPARX עם מוצר בשם Enterprise Architect(EA)
3. BORLAND עם מוצר בשם Together
4. ARIS
5. SELECT

התקן עבר מאז שינויים רבים וכיום המהדורה העדכנית של UML הינה 2.5

להלן כלי פיתוח ליצירת תרשימי UML 2.X אשר זמינים בחינם(בדרגת איכות נמוכה יותר):

כתובת	מוצר
http://staruml.sourceforge.net/en/download.php	Star UML
http://www.netbeans.org	Netbeans
http://www.eclipse.org	Eclipse
http://www.visual-paradigm.com	Visual Paradigm

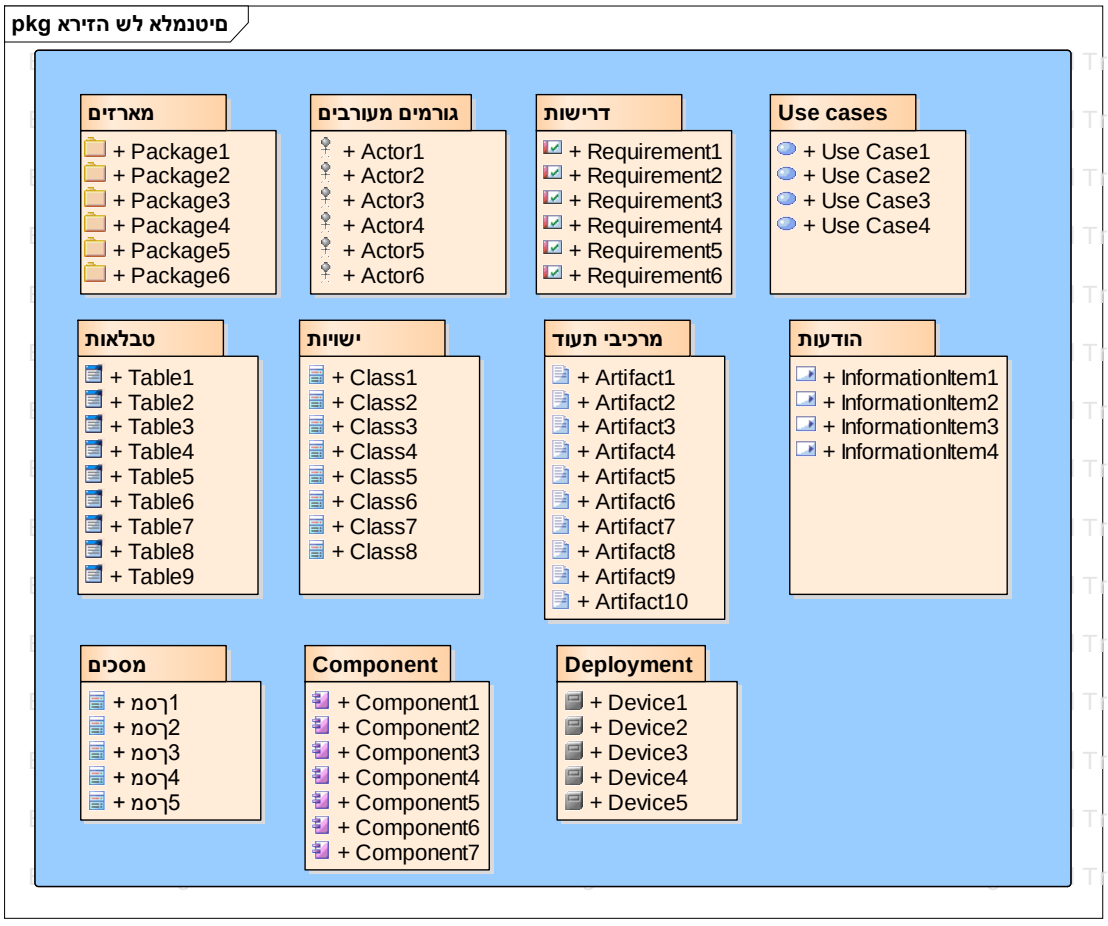
על מנת ליצור תרשימי UML ניתן להשתמש גם בכלים שאינם יעודיים רק ל UML לדוגמא:
VISIO, VISUAL STUDIO, Word, אקסל
הסברים מפורטים על כל דיאגרמה ניתן למצוא באתר ישראלי ללימוד UML

Structure Diagrams

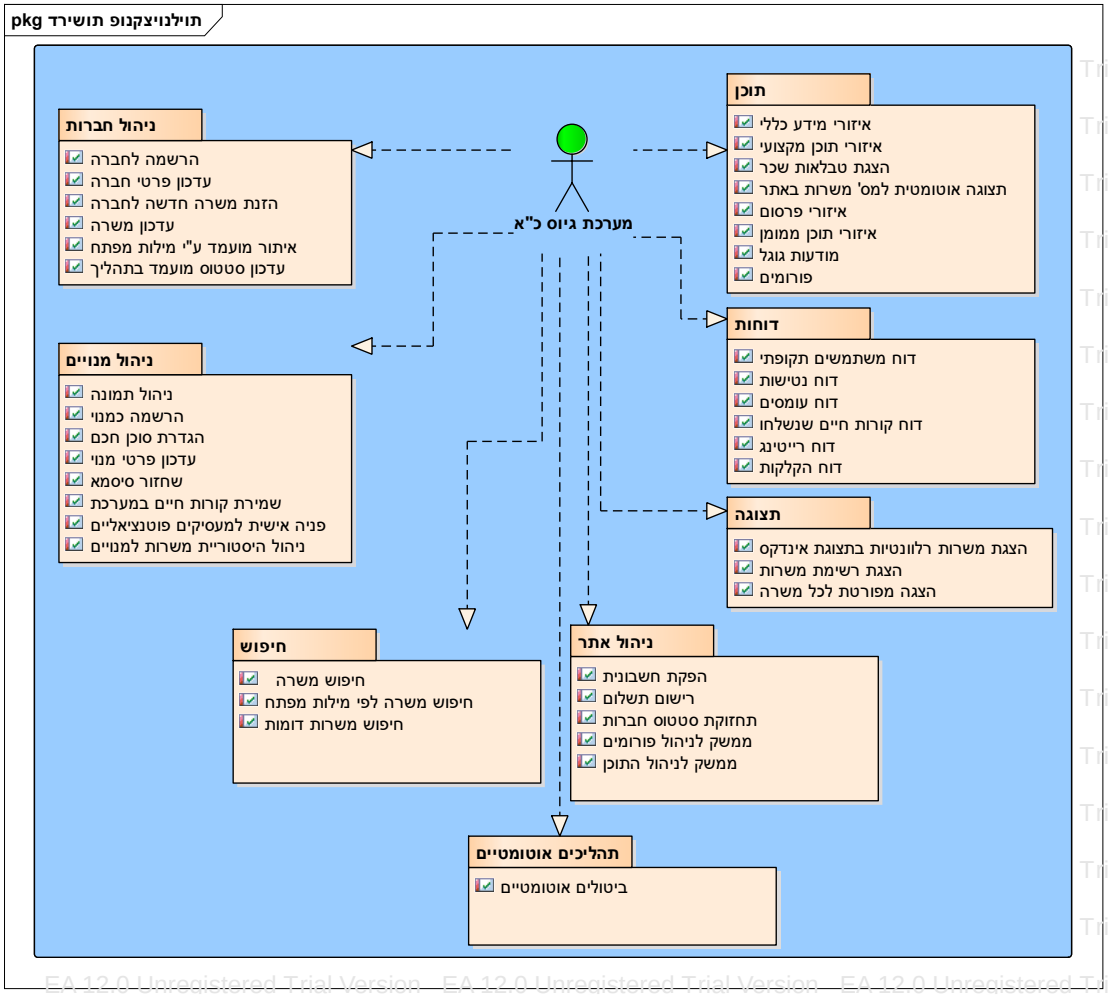
הדיאגרמות אשר שייכות לקבוצה זו מאפשרות להציג את המבנה הסטטי של המערכת, כלומר כל האלמנטים שהמערכת הממוכנת כוללת

Package Diagram-תרשים מארזים

- זהו תרשים שדומה במהותו לספריה ב Windows
- לתרשים אין שום משמעות לוגית מיוחדת , אבל הוא קריטי לארגון נכון של כל המידע על הפרויקט
- התרשים מאפשר לאגד בתוכו אוסף רצוי של אלמנטים
- לדוגמא: מארז של דרישות, מארז של מחלקות, מארז של תהליכים, מארז של קומפוננטות, מארז של מארזים וכו
- הרעיון המרכזי הינו לשמור על עיקרון ה 7±2 על מנת להקטין את הסיבוכיות. בכל מארז מומלץ לשמור 7+2 אלמנטים זהים
- להלן דוגמא של מארזים לסוגי אלמנטים שונים

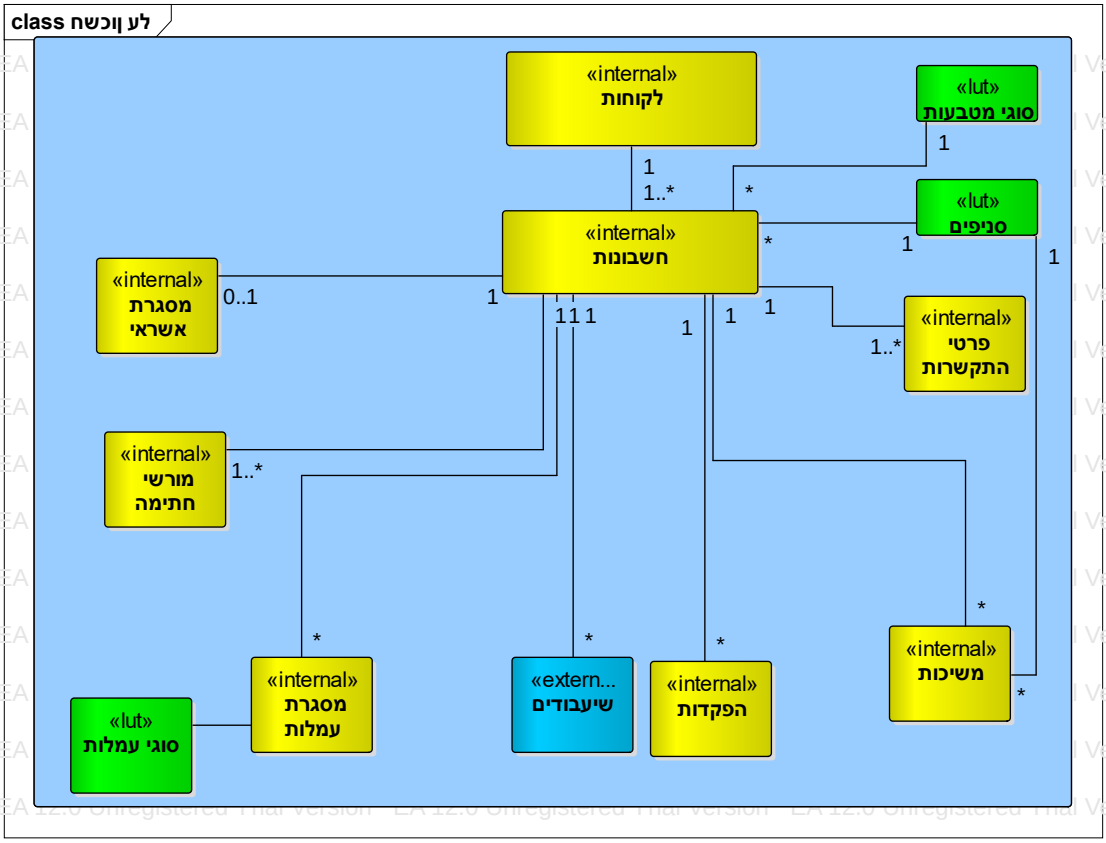


- להלן דוגמא של חלוקת הדרישות של פרויקט לאוסף של מארזים

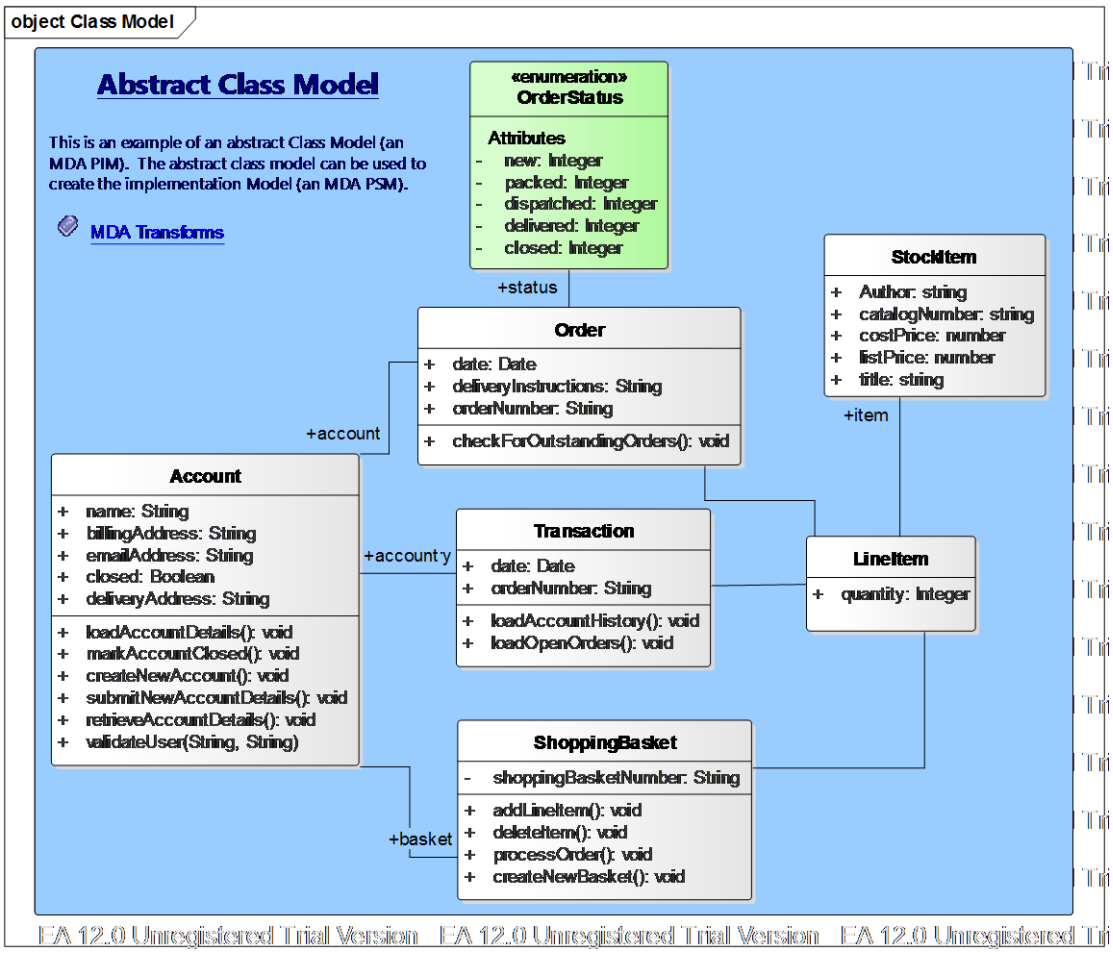


Class Diagrams

- כיון ששפות הפיתוח המודרניות(JAVA, C# ,Python ..) הן כולן מונחות אוביקטים גם UML מבוסס על אוריינטציה אוביקטלית ורב האלמנטים שלו תומכים במושג המחלקה
- מחלקה הינה אלמנט יסודי של UML. אחד השימושים העיקריים של מחלקות הינו לתאר את מודל הישויות בדרגות שונות של פירוט.
- באמצעות Class Diagram ניתן לתאר את המחלקות שאנו עומדים להגדיר, את הקשרים שיש ביניהן, את המאפיינים (ה attributes) ואת המתודות (ה operations)-שכל Class יכלול בהגדרתו.
- באמצעות Class Diagram ניתן להציג את המערכת המתוכננת מנקודת ראות סטטית בלבד.
- כל Class מתואר באמצעות מלבן אשר מחולק לשלושה חלקים. בחלק העליון רושמים את שם ה-Class, זאת דרגת הפירוט הראשונה
- בחלק האמצעי מפרטים את המאפיינים שיוגדרו בו , זוהי דרגת הפירוט השניה
- בחלק השלישי מגדירים את הפעולות(Operations) שכל מחלקה מסוגלת לבצע כשירות עבור מחלקות אחרות- דרגת פירוט זו משמשת בעיקר לעיצוב המערכת
- להלן דוגמא של דרגת הפירוט הראשונה-תרשים על של מודל ישויות לוגי



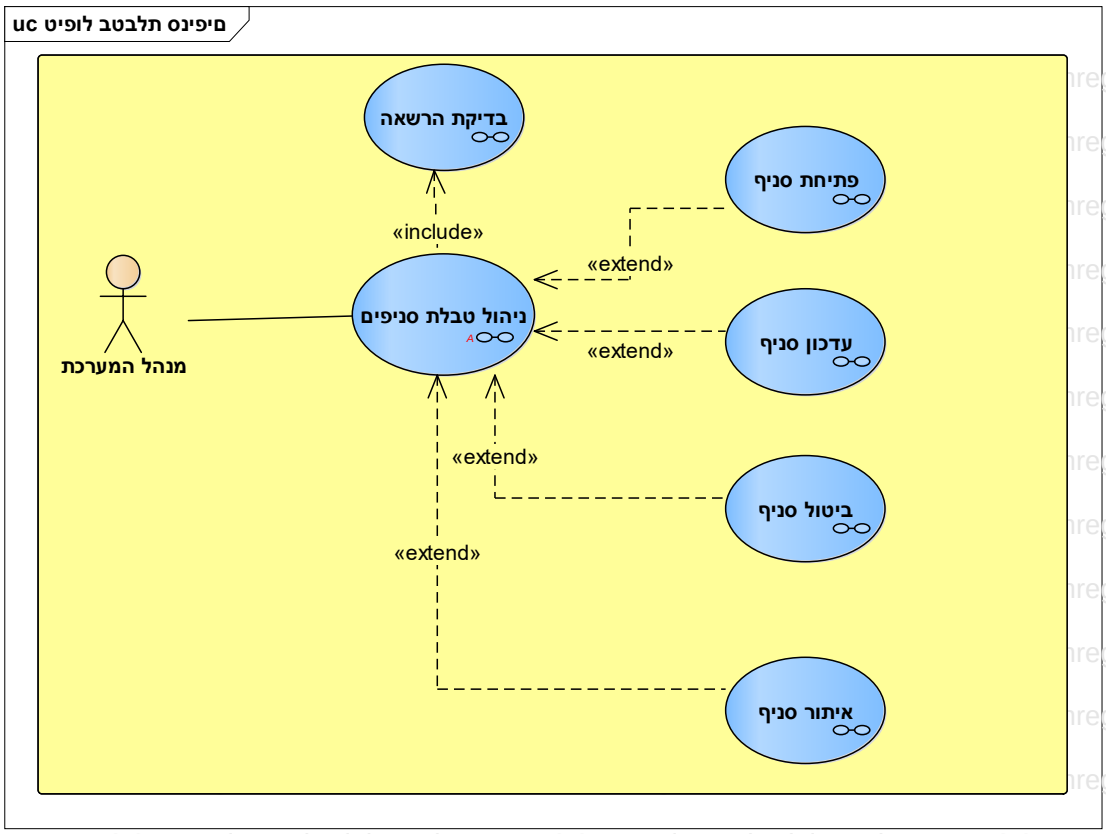
- להלן דוגמא של דרגת פירוט שניה
- להלן דוגמא של דרגת פירוט שלישית- בדרגת פירוט זאת מומלץ להשתמש באנגלית כדי לאפשר יצירה אוטומטית של קוד.



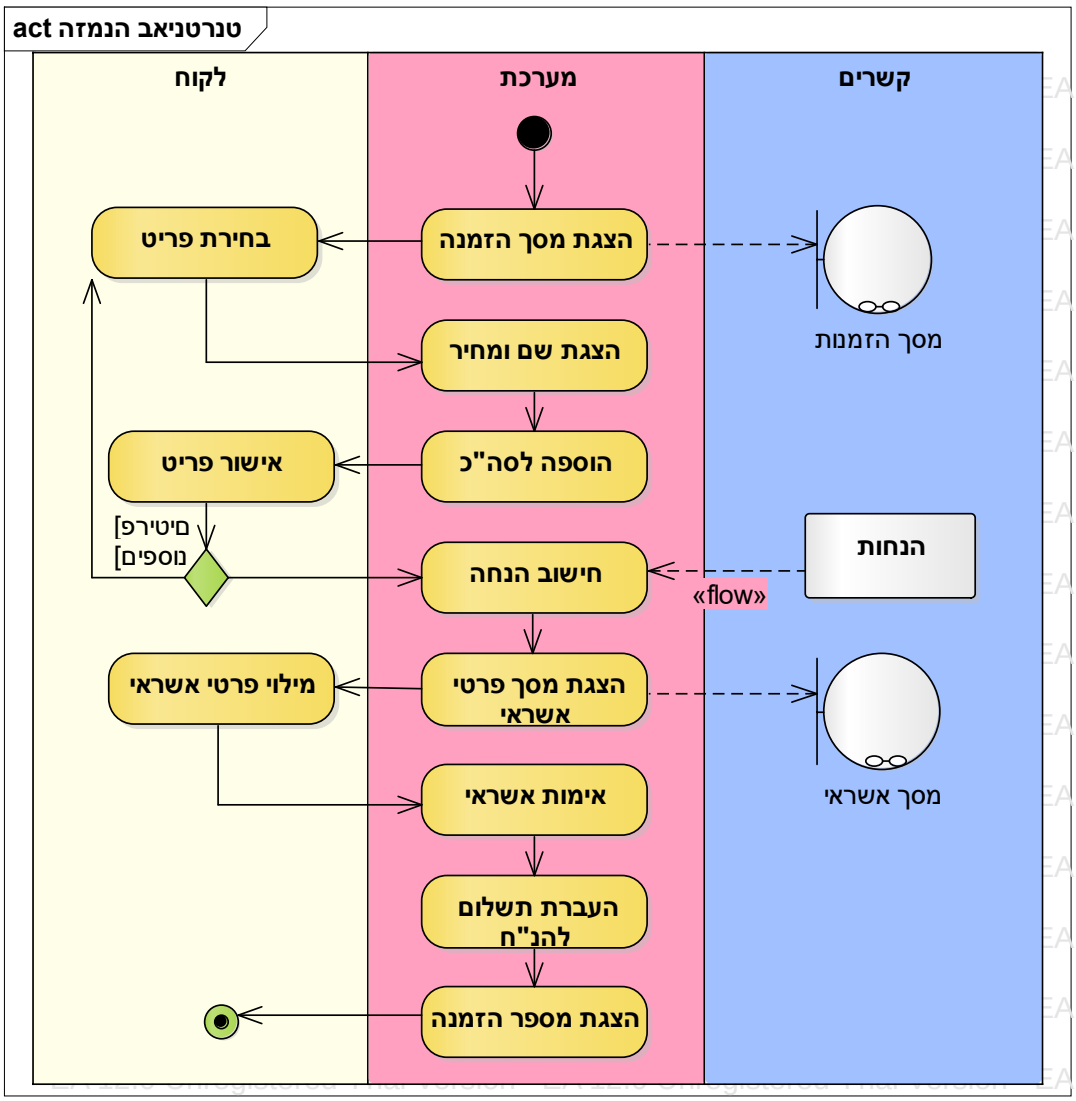
Behaviour Diagrams

דיאגרמות התנהגותיות

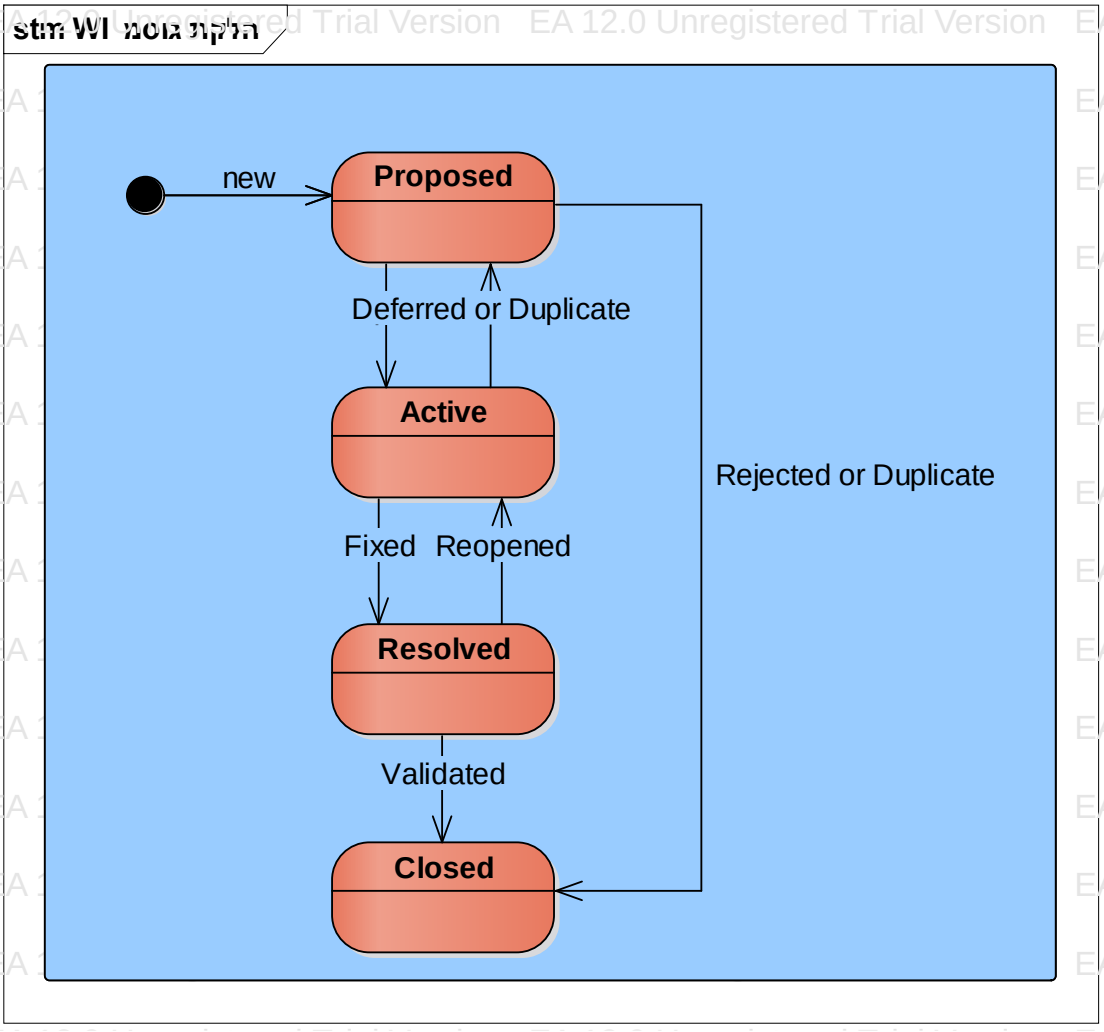
- קבוצת דיאגרמות זו מאפשר להציג מודל דינמי של המערכת, דהיינו האינטראקציות בין האלמנטים השונים
- תרשימי ה Use Cases(משימות) מציגים את הפונקציונאליות של המערכת , כלומר המשימות השונות שהשחקנים יכולים לבצע ואת קשרי התלות בין המשימות השונות.התרשימים מצגיים את הפונקציונאליות העסקית של המערכת מנקודת המבט של המשתמש, להלן דוגמא אופיינית



- תרשימי Activity מפרטים את הפעילויות השונות ,צעד אחר צעד, כולל סדר הפעולות וההתניות הדרושות לביצוע משימה, תרשים זה הינו גלגול מודרני של תרשים הזרימה. להלן דוגמא אופינית


טנרטניאב הנמזה act
לקוח
מערכת
קשרים
בחירת פריט
הצגת מסך הזמנה
מסך הזמנות
הצגת שם ומחיר
אישור פריט
הוספה לסה"כ
[מיטירפ נוספים]
הנחות
חישוב הנחה
«flow»
מילוי פרטי אשראי
הצגת מסך פרטי אשראי
מסך אשראי
אימות אשראי
העברת תשלום להנ"ח
הצגת מספר הזמנה

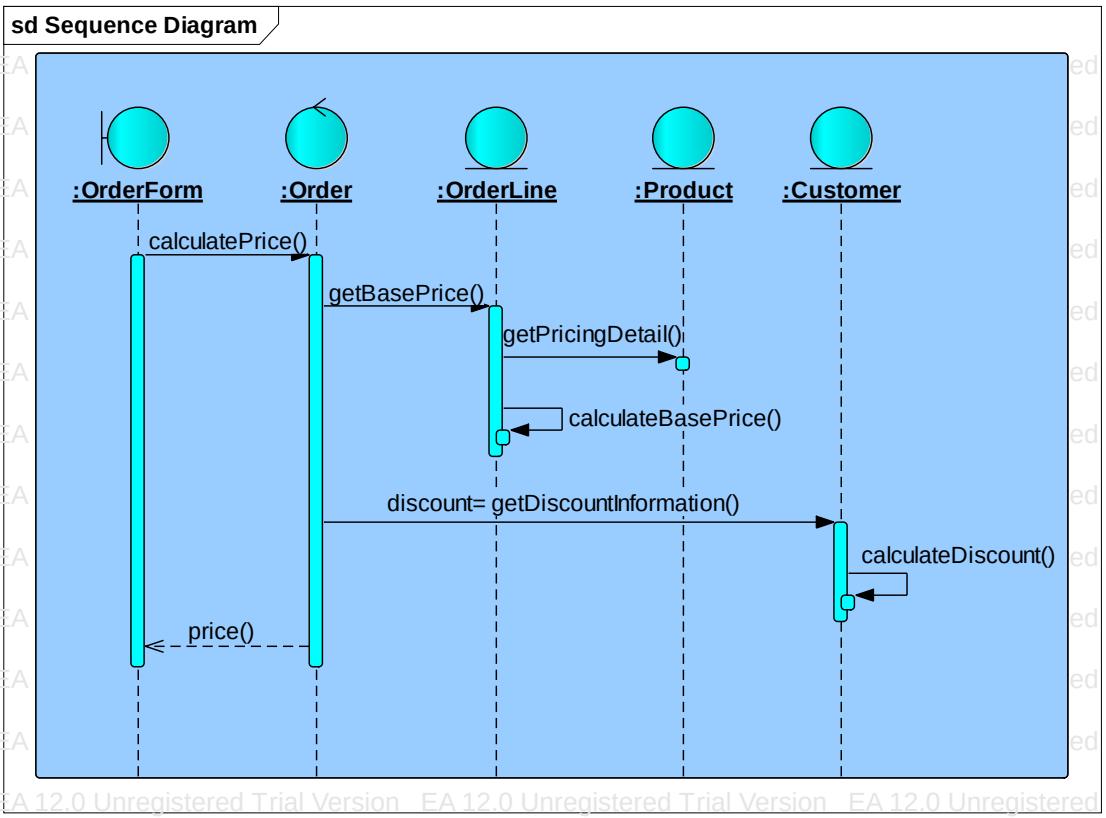
- תרשימי מצב מאפשרים להציג את מחזור החיים של ישות על ידי הצגת המשימות השונות שמעבירות את הישות מסטטוס לסטטוס, להלן דוגמא אופינית.



Sequence Diagrams

תרשימי רצף

- תרשימים אלו נועדו לפרט Use Cases ברמת פירוט שנועדה למפתחים.
- התרשים מפרט את המסרים שעוברים בין האובייקטים השונים לצורך ביצוע פעולה רצויה
- להלן דוגמא אופינית


sd Sequence Diagram
:OrderForm
:Order
:OrderLine
:Product
:Customer
calculatePrice()
getBasePrice()
getPricingDetail()
calculateBasePrice()
discount= getDiscountInformation()
calculateDiscount()
price()
EA 12.0 Unregistered Trial Version
EA 12.0 Unregistered Trial Version
EA 12.0 Unregistered