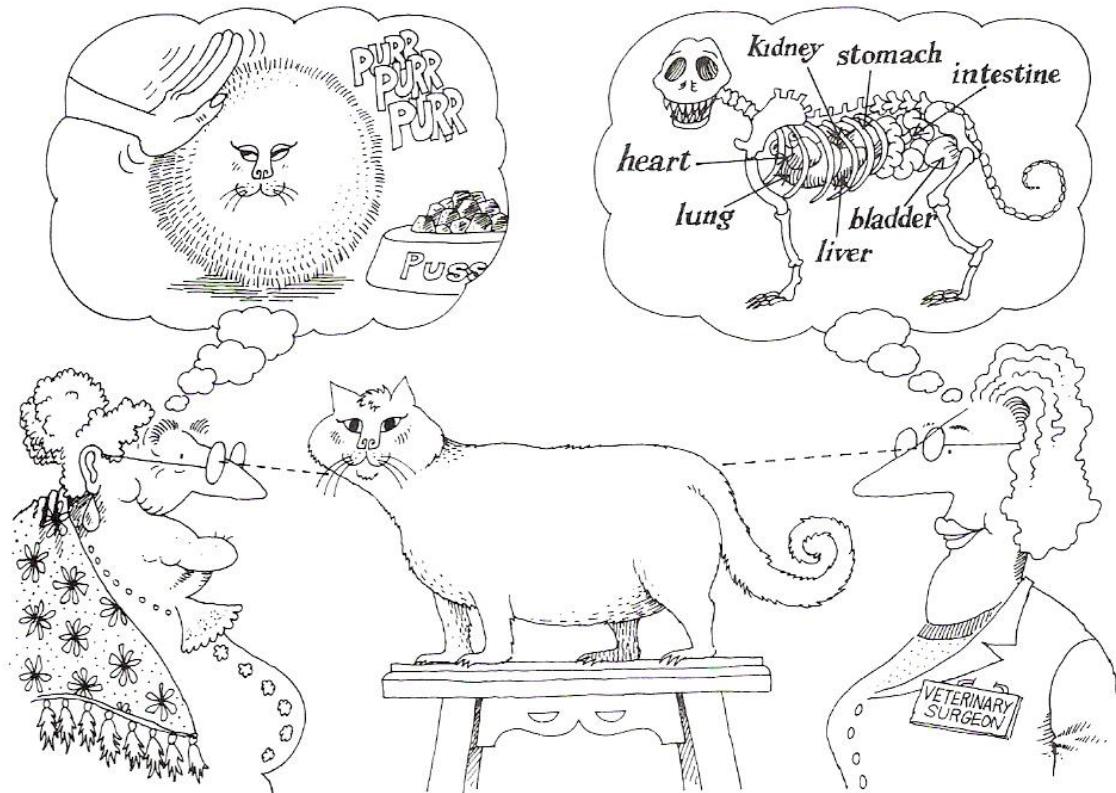


Requirements Analysis and Specification



Picture Source: Booch, G., *Object-Oriented Analysis and Design*, Benjamin/Cummings, 1994

- תרשימים Use Cases
- תרשימים Use Cases בرمאה ארגונית
- תרשימים Use Cases בرمאה מערכתית
- Use Case
- שחקן Actor
- שחקן ראשי
- שחקן משני
- יחסים בין UC

שירות המעליות – שלוש הרמות העליונות

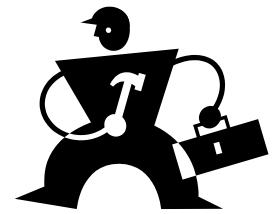
רמת הארגון
Business Level

רמת המערכת
System Level

רמת המרכיב/התוכנה
Software Level

מחלקה האחזקה של ביה"ח (שירות המעליות)

טכני מעליות – איש חילוץ
לודיעתך המפתח לחדר המכוון
של המעלית נמצא אצל
מנהל התחנה או אצל איש הביטחון
תודה.



טכני

מערכת המעליות

מעליות

פיקוד
מרכזי

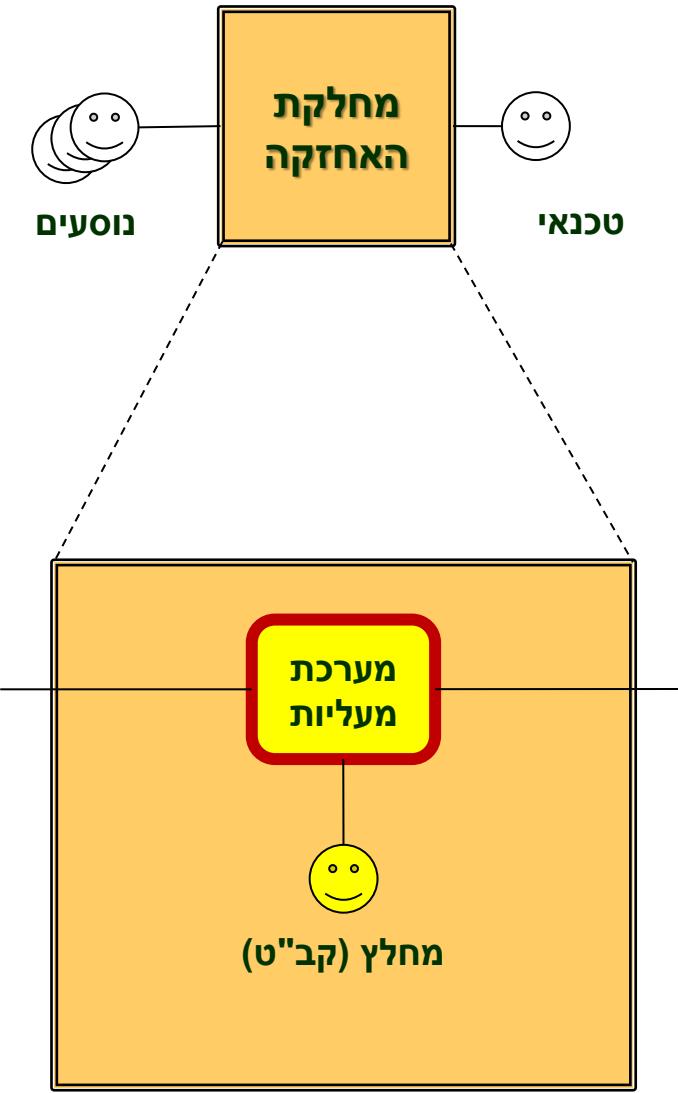
נוסע



מלחץ



רמת הארגון/העסק (בקשר של מערכת המעליות)



- הסביבה החיצונית
 - נוסעים
 - טכני Achzaka
- שירותים
 - הסעה בין קומות
 - חילוץ נוסעים תקועים
 - תחזוקת מערכת
- ממשקים
 - פאנל טכני
 - פאנלים (כפטורים) לנוסע
- מבנה
 - מערכת מעליות
 - מחלץ אנושי + פאנל חילוץ
- התנהגות
 - לוגיקת פעולה מעליית
 - לוגיקת חילוץ
 - לוגיקת ביצוע תחזוקה

מארגן דרישות לעומת מודל דרישות

- מאגר דרישות הינו רשימה/טבלה/בסיס-נתונים של דרישות בדידות, המסוגות על פי קטגוריות /או מאורגנות בקבוצות

– מאגר הדרישות משמש ...

- **רשימת-тиוג (check-list) לכיסוי התכן/הIMPLEMENTATION**

- **רשימת תיוג לאימות ולתיקוף**

- **בסיס להקצאת דרישות לרכיבי המערכת**

– דוגמאות: DOORS, טבלאות אקסל, ...

- **מודל דרישות פונקציונליות (תפעוליות)** מהוות ייצוג של הפעולות והאינטראקטיות של המערכת, על בסיס הדרישות התפעוליות הבודידות

– **מודל הדרישות התפעוליות** משמש לתקן ולימוש התנהגות המערכת

- **התנהגות המערכת ממומשת במקדים רבים בתוכנה**

דרישות פעולה ומודל Use Cases

- מודל Use-Cases נועד לספק מפרט שלם להנהגות המערכת על בסיס **הדרישות הפעוליות**
 - הדרישות הפעוליות מקובצות לתרחישים פעולהים
 - מוגדרים היבטים נוספים של ההקשר הפעולי, כגון
 - התנאים המתקיים לפני ואחרי ביצוע תרחיש כלשהו
 - האינטראקציה עם הסביבה החיצונית
 - האינטרסים של בעלי העניין בכל הקשור לתפעול המערכת
- המודל מתיחס לסוגים אחרים של דרישות ככל שהן רלוונטיות לתפעול המערכת
- מודל ה-Use-Cases נועד לתת כיסוי מלא לדרישות הפעוליות של המערכת
 - תוך כדי יישוב קונפליקטים, חוסרים וכו'

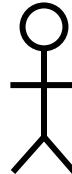
- מtower כלל בעלי העניין בפיתוח,

בעל עניין במערכת (System Stakeholder) הינו אדם, צוות או ארגון אשר יש לו אינטרסים או נושאי עניין (concerns) הנוגעים **לפעולת המערכת**

- השירותים אותו הוא מczפה לקבל ממנו, או שהוא מצופה לספק לה
- ביצוע פועלות נוספות לצרכיו, גם תוך כדי מתן שירותים לבעלי עניין אחרים
- aicות תפעול המערכת תוך מתן שירותיה
- השפעת התפעול של המערכת עליו, תוך כדי פעולה ולאחראית
- אופן עמידת תפעול המערכת בסטנדרטים/רגולציות אותם הוא מייצג
- ...

- **שחקן (Actor)** הינו ישות (אנושית או לא-אנושית) **בסביבה החיצונית** של המערכת, המסוגלת לבצע **אינטראקטיה ישירה** אליה
 - שחקנים הינם בעלי עניין במערכת, או נציגים של בעלי עניין במערכת
 - **למשל, מחשב חיצוני הנמצא באינטראקטיה עם המערכת איננו בעלי עניין בפני עצמו כי אין לו אינטרסים משלו בפועלתה, אך הוא מייצג את האינטרסים של בעל(י) עניין ("אמתיהם")**
 - לא כל בעלי העניין במערכת הינם שחקנים!
 - התקני ממושך- משתמש (IMH), כגון מקלדת, עכבר, מיקרופון, קורא-כרטיסים וכו', ניתן לראות כחלק בלתי נפרד מהשחקן עצמו
 - **כלומר, למרות שהמשתמש האנושי אינו מחובר ישירות למערכת, הוא השחקן, ולא המקלדת**

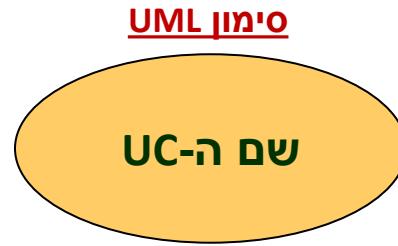
סימון IMU

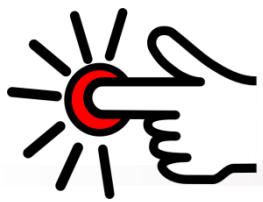


שם השחקן

Use Cases

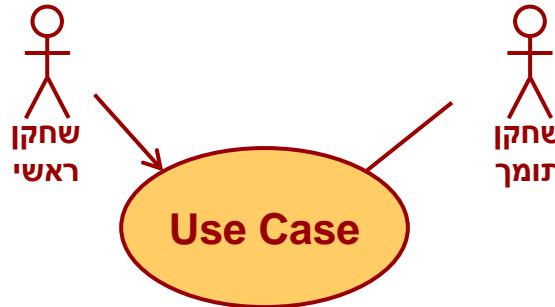
- **Use Case (UC)** הינו מטלה **שלמה** המבוצעת על ידי המערכת מתוך **מטרה** להפיק **תוצאות מוגדרות וبنות-מדידה** עבור **בעל-ענין** אחד או יותר
 - UC יכול להתבצע ביוזמת שחקן או ביוזמת המערכת עצמה
- UC המתבצע ביוזמת המערכת, כתוצאה מאירוע או תנאי בתוכה, לטובת בעל עניין במערכת. **לא כל פעולה יזומה על ידי המערכת תמודל במודל UC.**
- UC הוא תיאור מוסכם של אינטראקציה בין המערכת לשחקניתה
 - האינטראקציה /או תוצאות **נראות (visible)** לבעל העניין
- UC כולל את כל האינטראקציות האפשריות שנitinן לבצע לצורך השגת אותן תוצאות, **בין אם תוצאות אלה הושגו ובין אם לאו**
- מספר מופעים של אותו UC יכולים להתבצע במקביל





Use Cases ופתרונות

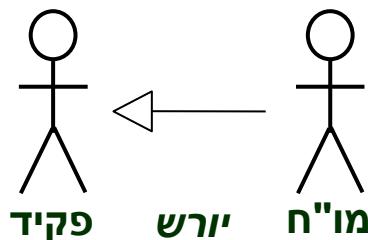
- כל UC משוויר לקבוצת שחקנים, משנה סוגים:
 - **שחקן ראשי (Primary Actor)** הינו שחקן היכול **לייזום** (להפעיל) ביצוע של UC
 - הייזום נעשה דרך אחד **המשקיעים המספקים*** של המערכת
 - כאשר יש יותר שחקן ראשי אחד המשמעות היא **שכל אחד** מהם יכול לייזום את ביצוע-UC באופן **בלתי תלוי**
 - שחקן ראשי יוזם ביצוע של UC כדי להשיג **יעד** כלשהו
 - **שחקן תומך (Supporting Actor)** הינו שחקן (שאינו ראשי) אשר המערכת מבצעת אליו אינטראקציה תוך כדי ביצוע של UC
 - האינטראקציה מתבצעת ביוזמת המערכת דרך אחד **המשקיעים הנדרשים*** שלו
 - תפקידו של שחקן תומך הוא **לסייע** למערכת להשיג את מטרתה



* המונחים **שחקן מספק** ומ**שחקן נדרש** יוגדרו בהמשך הקורס

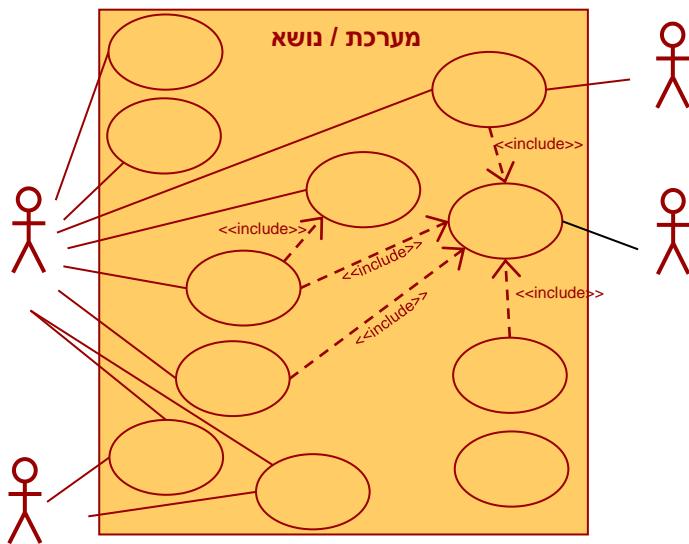
עוד על Use Cases ושחקנים

- UC ללא שחקנים ראשיים הוא UC שהמערכת יוזמת.
- שחקן יכול "לרשת" שחקן אחר
 - כאשר שחקן A יורש שחקן B הכוונה היא ש-A יכול לעשות כל מה ש-B יכול, בתוספת לפעולות הייחודיות לו בלבד
- במידול מונחה עצמים לכל UC יהיה primary actor אחד.
- פרט לשחקנים, משוייך לכל UC לקבוצה של בעלי עניין רלוונטיים
 - **בעל עניין רלוונטי** הינו בעל עניין של המערכת שיש לו **אינטרס(ים) ספציפי(ים)** בביצוע של UC זה



תרשים (Use Case Diagram) Use Cases

- תרשים Use Case (UCD) הינו ייצוג גרافي של קבוצת Use Cases וקשריהם עם קבוצת שחקנים
 - למרות הקשר הדוק בין Use Cases ובין בעלי עניין שאינם שחקנים, אין ב-UML ייצוג גרافي שלהם
- התרשים כולל את המרכיבים הבאים:
 - מלבן תוחם, המציג את גבולות מערכת-העניין / הנושא / Subject)
 - אליפסות בתחום הגבולות, המציגות Use Cases
 - דמות מחוץ לגבולות, המציגות שחקנים
 - כל שחקן מקשר ל-UC אחד או יותר
 - חיצים מקווקויים, המציגים יחסים בין UCs
 - יוסברו בהמשך



למרות שתתרשים ה-UC נועד לייצג את **תפועל המערכת**, הוא עצמו אינו מודל דינמי!

– מרכיבי המודל – Use Case Model

- Use Case Diagram
 - פירוט השירותים והקשרים בין השחקנים
- Use Case Specification
 - תיאור של אופן קבלת השירות, כאינטראקציה של השחקנים עם המערכת
 - בטקסט (מבנה) או במודול דינמי אחר
- עקיבות (traceability) דו-כיוונית לדרישות התפעוליות
 - בטבלת הדרישות יש הצבעה מכל דרישת פעולה ל-UC המממש אותה
 - מכל Use Case יש הצבעה לדרישות התפעוליות אותן הוא מממש

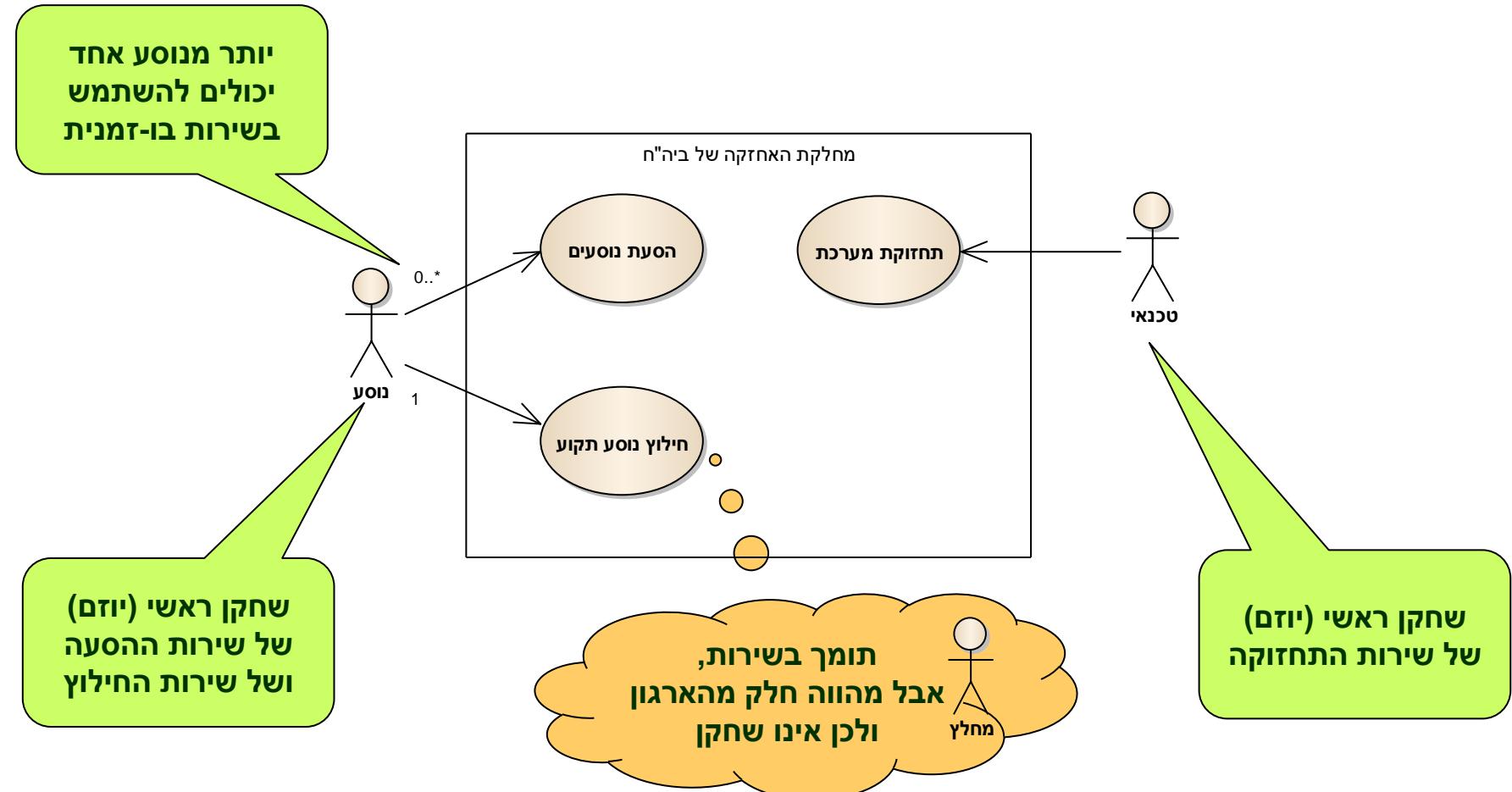


Use Case Model
אמור
לכסות את כל הדרישות
התפעוליות (OR) בטבלת
הדרישות

מחלקה האחזקתית (שירות המעליות) – בעלי עניין ואינטרסים כלליים

תפקיד	אינטרסים כלליים [תועלות, מאפייני איזות]	בעל עניין
משתמש	העה מקומה לקומה באופן מהיר ובטוח [שימושיות, ביצועים, אמינות, זמינות]	נוסף
משתמש	ביצועiesel ואמין של תחזקה [בדיקות]	טכני
מבצע	חילוץ אמין ובטוח של נוסעים שנתקעו [בטיחות]	מחלץ
-	אמינות ובטיחות	תקנים
-	בדיקות, בטיחות, אמינות	משרד העבודה

מחלקה האחזקה – Use Case Diagram



חסמים בין Use Cases: הכללה - Generalization

- מקרה השימוש המהווה בן ירוש את התנהגות והמשמעות של מקרה השימוש המהווה אב.
- הבן יכול להוסיף או לדרכו את התנהגות האב.

הוראה

כג

רישום

רישום
מוסמכים

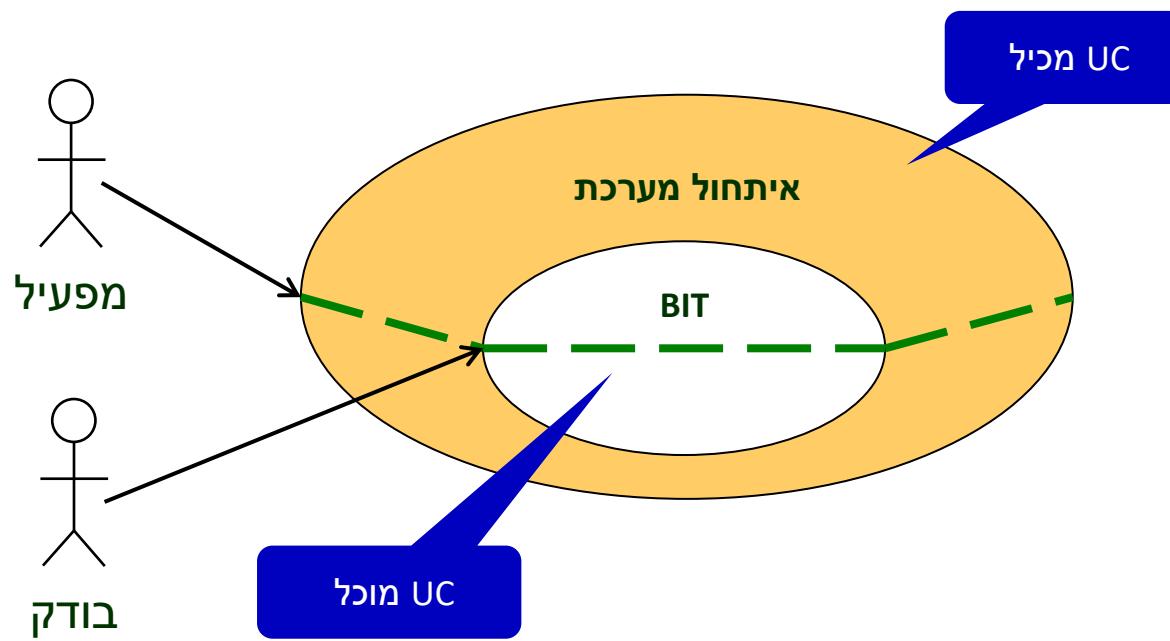
רישום
לא מוסמכים

סימול:

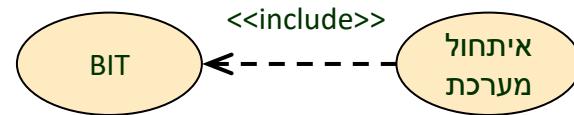
דוגמה:

תלות "הכלה" («include») בין Use Cases

- Use Case B הוא **מכלול** Use Case A
 - B עדין יכול להיות מופעל באופן עצמאי, ע"י שחקנים אחרים
- **לדוגמה** BIT (Built in Test)



סימן UML

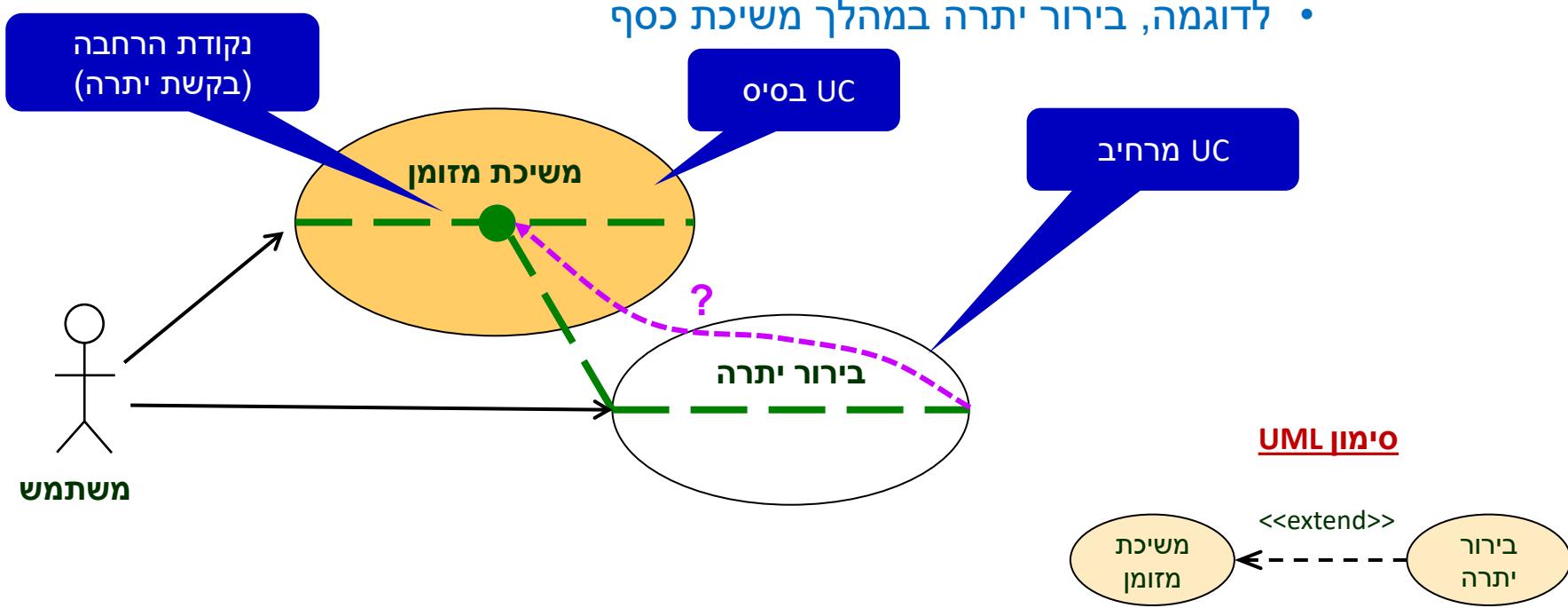


מתי להשתמש ב- <<include>> ?

- כדאי להפריד חלק מ-UC ולהגדירו כ-UC עצמאי באחד או יותר מהמקרים הבאים:
 1. ה-UC המוכל הוא גדול (כול מספר רב של פעולות)
 2. ההתנהגות ה"מוכלת" משותפת ליותר מ-UC אחד (ואז היא תהיה מוכלת בCOLUMN)
 3. ניתן להפעיל את ה-UC המוכל בנפרד מ-UC הבסיס

תלות "הרחבה" («extend») בין Use Cases

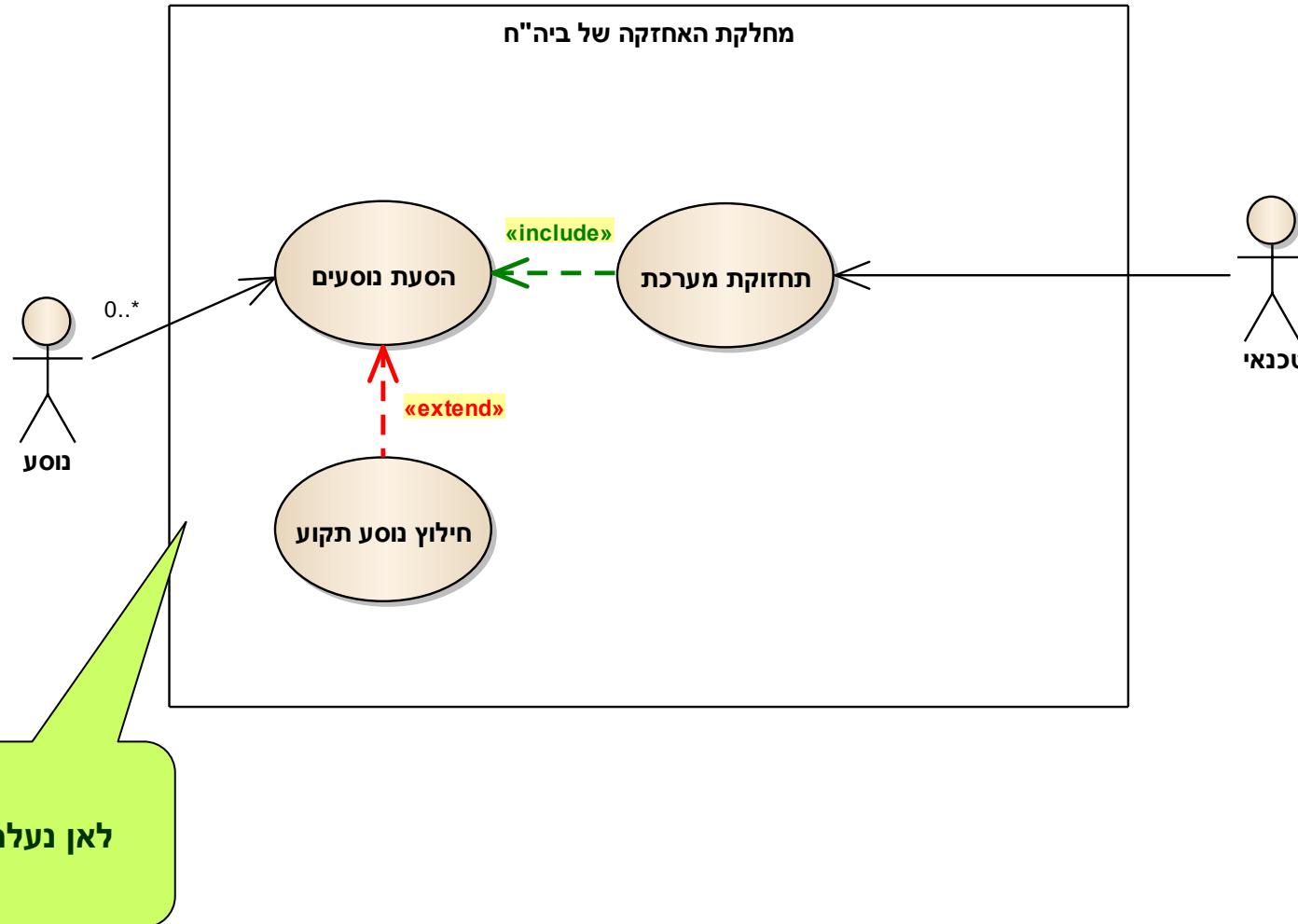
- Use Case A מרחיב את Use Case B ניתן להפעלה אופציונאלית במהלך ביצוע A
 - B מופעל ב"נקודות הרחבה" (extension point) מוגדרת בתוך A
 - B עדין יכול להיות מופעל כ-UC עצמאי ע"י שחקנים אחרים



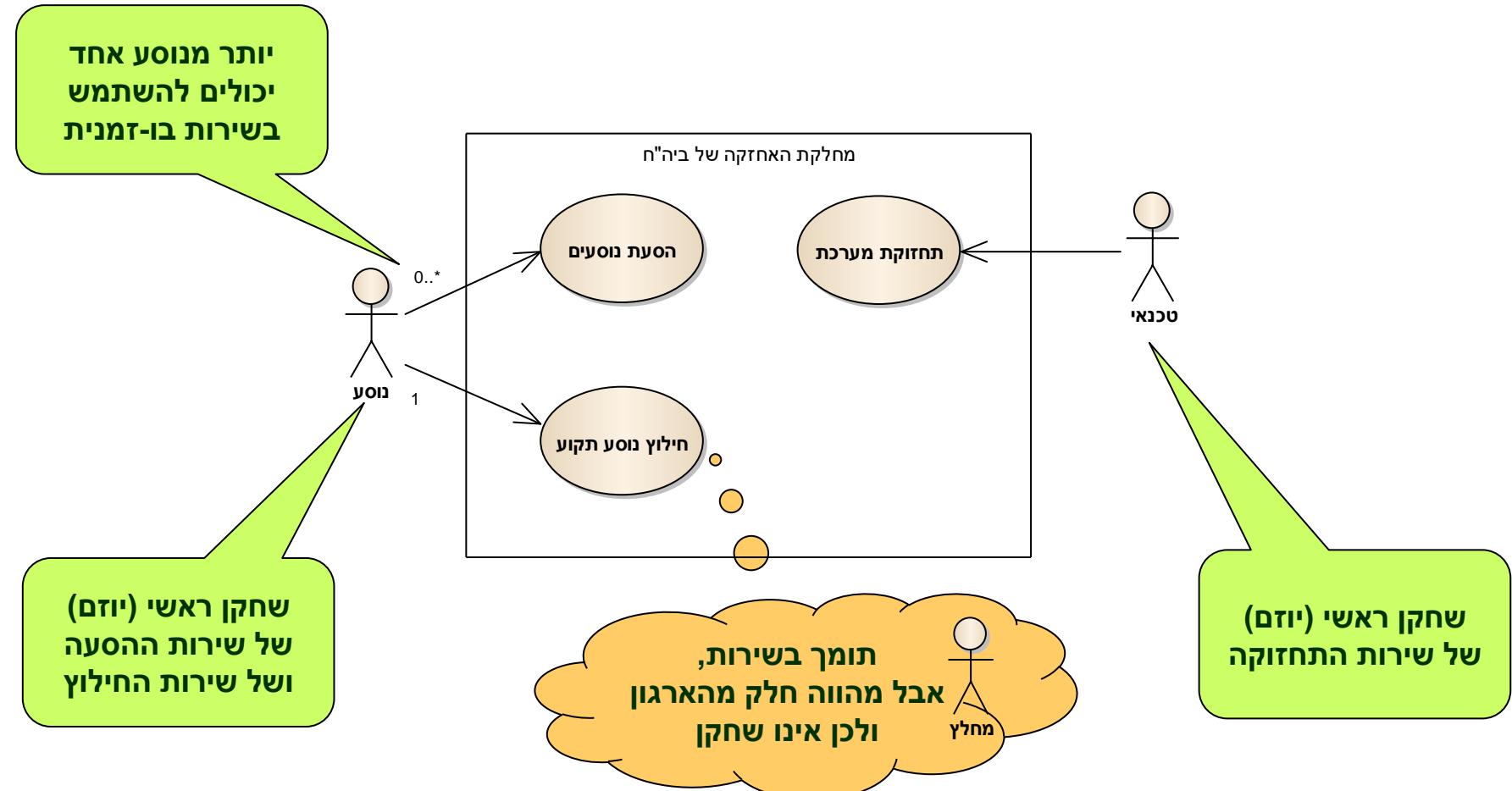
מתי להשתמש ב- <>extend> ?

- ברוב המקרים ניתן להשיג את אותו אפקט ע"י תרחיש הסתעפות בתוך UC הבסיס (יפורט בהמשך)
- כדאי להפריד את התנהגות UC הבסיס ל-UC מרחיב נפרד במקרים הבאים:
 1. ההתנהגות המרחיבה היא גדולה (מכילה פעולות רבות)
 2. ההתנהגות המרחיבה משותפת ליותר מ-UC בסיס אחד
 3. ההתנהגות המרחיבה ניתנת להפעלה עצמאית

מחלקה האחזקה (שירות המעליות) – תרשימים עם תלויות



מחלקה האחזקה – Use Case Diagram



שירות המעליות – מפרט Cases בرمת הארגון

- **מפורט לא פורמלי** (פירוט האינטראקציה בין נותן השירות ומקבל השירות)
 - **הסעת נוסעים** (על בסיס סיפור הלוקוח)
 - הנושא מזמין מעליית לקומת המוצא, המעלית מגיעה, הדלת נפתחת והנוסע נכנס לתוכה
 - הנושא מזין קומת יעד, נוסע במעלית לקומת היעד, המעלית נעצרת, הדלת נפתחת והנוסע יוצא
 - **חילוץ נוסע תקוע** (על בסיס סיפור הלוקוח)
 - הנושא מזעיק חילוץ, מחלץ מגיע, מוריד את המעלית לקומת הקרקע ומחלץ את הנוסע
 - **תחזוקת מערכת** (לא מפורט בסיפור הלוקוח)
 - טכני מבצע בדיקה למערכת
 - במקרה שהתגלתה תקלת הטכני מתקן אותה ומבצע בדיקה חוזרת

מטלת כתה

- על בסיס סיפור הלקוח:
 1. זהה והגדיר את "הארגון" ואת "המערכת"
 2. זהה את בעלי העניין (חיצוניים לארגון), את האינטרסים שלהם ואת תפקידיהם
 3. זהה את מרכיבי הארגון (פנימיים לארגון), את ייעודיהם ואת תפקידיהם
 4. זהה Use Cases ברמת הארגון (שירותים שהארגון מספק לבעלי העניין שלו)
 5. עורך Use Case Diagram המכיל את כל היחסות הנ"ל ואת הקשרים שביניהם.

מערכת מעליות – סיפור הלקוּח (ובעליו העניין) : תיאורי תהליכיים

נוסע הנמצא בקומה כלשהי ורוצה לנסוע במעלה לוחץ על הכפטור המתאים לכיוון הנסיעה המבוקש. אם לא היה דלوك קודם לכך, נדלק הכפטור בעקבות הלחיצה. מעלית כלשהי הנמצאת בכיוון הנסעה המבוקש תגיע לקומתך, תוך דקה לכל היתר. עם הגעתה תיפתח הדלת והכפטור יכבה.

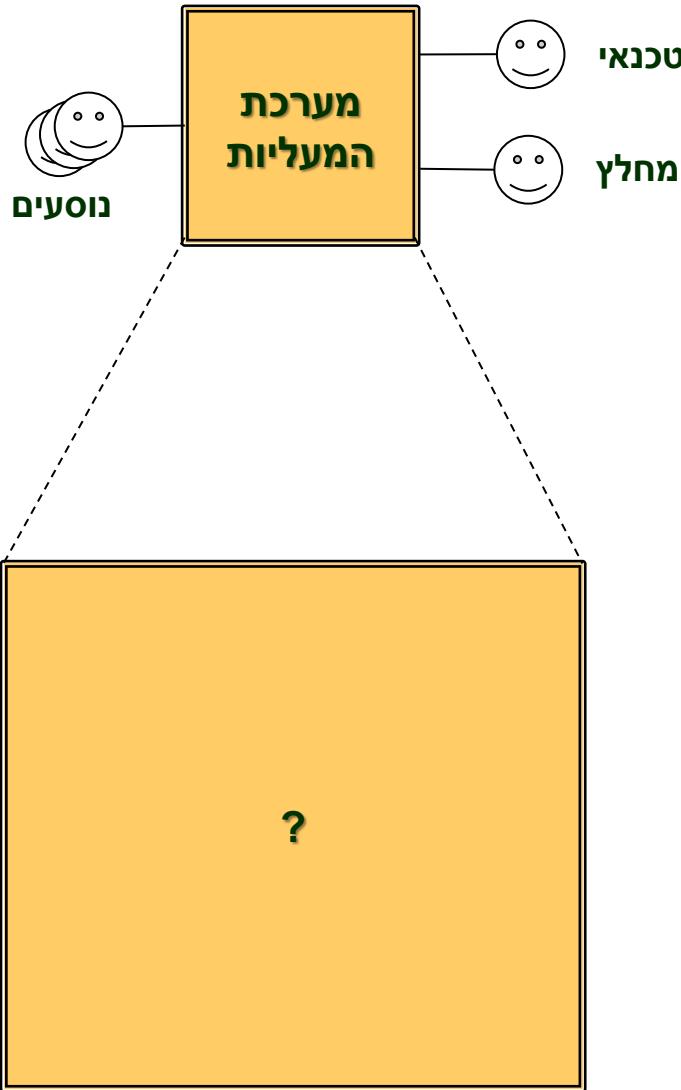
בתוך כל מעלית נמצאים 10 כפטורים – אחד עבור כל קומה. כמו כן נמצאים במעלה כפטור לעצירת חירום וכפטור להזעקת חילוץ.

נוסע הנמצא בתוך המעלית ורוצה להגיע לקומת כלשהי לוחץ על הכפטור המתאים. אם לא היה דלוק קודם נדלק הכפטור בעקבות הלחיצה ולמעלית נוספת נסافت בקשה לעצירה בקומה המתאימה. כאשר תגיע המעלית לך קומה המבוקשת היא תיעזר, הדלת תפתח והכפטור המתאים לך קומה יכבה. לאחר שההיה תיסגר הדלת והמעלית תמשיך בפעולתה.

במקרה שהמעלית נתקעה במהלך נסעה מזעיק הנושא חילוץ באמצעות כפטור החילוץ. איש האחזקה של הבניין הוכשר לחילוץ נוסעים תקועים באמצעות פאנל הפעלה מיוחד הנמצא בחדר המכוון. **המחלץ מוריד את המעלית לקומת הקרקע ופותח את דלתותיה.**

על פי תקנות משרד העבודה נבדקת המערכת אחת לששה חודשים בידי טכנאי מוסמך. בזמן הבדיקה לא ניתן להשתמש במערכת. **אם הטכנאי מגלה תקלת הוא מנסה לתקן אותה ולהזoor על הבדיקה.**

רמת מערכת המחשב – Computer System Level



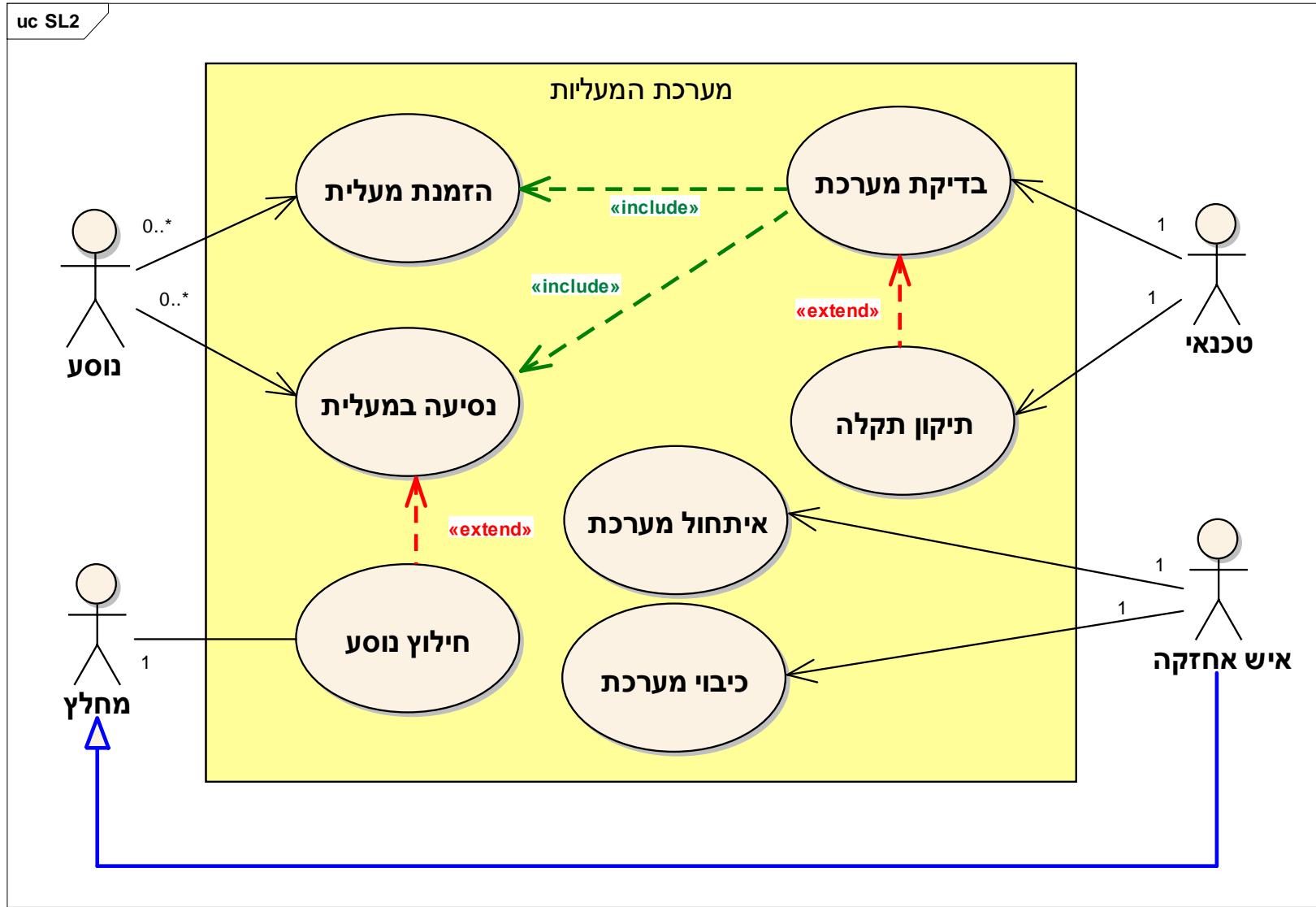
- הסביבה החיצונית
 - נוסעים
 - טכני
 - מחלץ
- שירותים
 - התהליכים המערכתיים הממשיכים את שירותי הארגון
 - תוצר תהליכי הניתוח המערכתית
- ממשקים
 - פאנל נוסע
 - פאנל טכני
 - פאנל מחלץ
- מבנה
 - ארכיטקטורת המערכת
 - תוצר תהליכי התקן המערכתית
- התנהגות
 - מימוש תהליכי המערכת באמצעות מרכיביה
 - תוצר תהליכי התקן המערכתית

מערכת המעליות – גזירת תהליכי המערכת משירותי הארגון

- על בסיס סיפור הלקווח / פירוט השירותים ברמת הארגון (מחלקת האחזקה)
 - שירות ההסעה מורכב מהתהליכיים הבאים
 - הנושא מזמן מעלית לקומת המוצא, המעלית מגיעה, הדלת נפתחת והנוסע נכנס לתוכה
 - הנושא נוסע במעלית לקומת היעד, המעלית נעצרת, הדלת נפתחת והנוסע יצא
 - שירות החילוץ מורכב מהתהליך הבא
 - במהלך וסיום נוסע מזעיק חילוץ, מחלץ מגיע, מוריד את המעלית לקומת הקרקע ומחלץ את הנוסעים
 - שירות התחזוקה מורכב מהתהליכיים הבאים
 - טכני מבצע בדיקה למערכת
 - במקרה שהתגלתה תקלת הטכני מתקן אותה ומבצע בדיקה חוזרת
- על בסיס ידע הנדסי ומאפייני אינטוף
 - תהליכי ניהול מערכת
 - איתחול המערכת (start up)
 - כיבוי המערכת (shut down)

מודדים ל-Use Cases ברמת המערכת

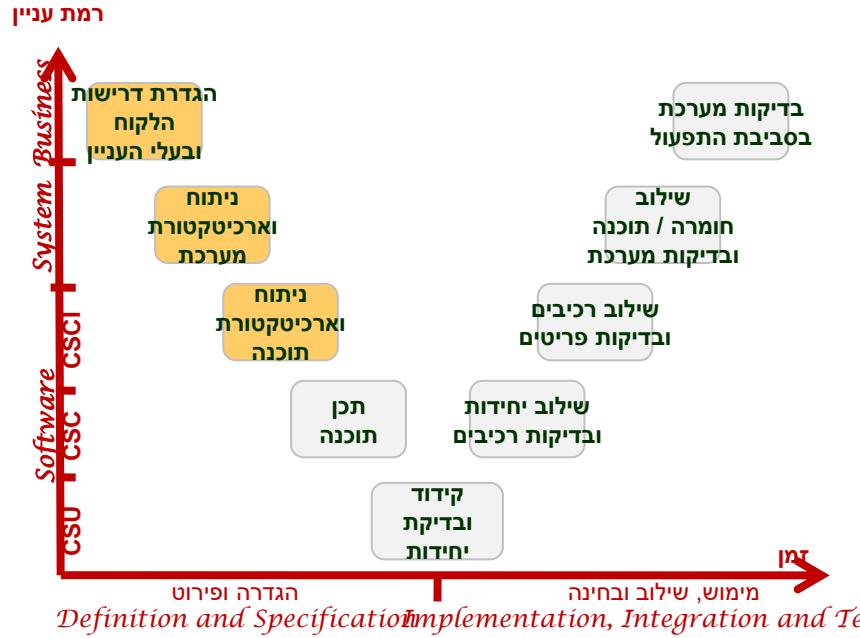
מערכת המעליות – System Level UC Diagram



מטלת כתה

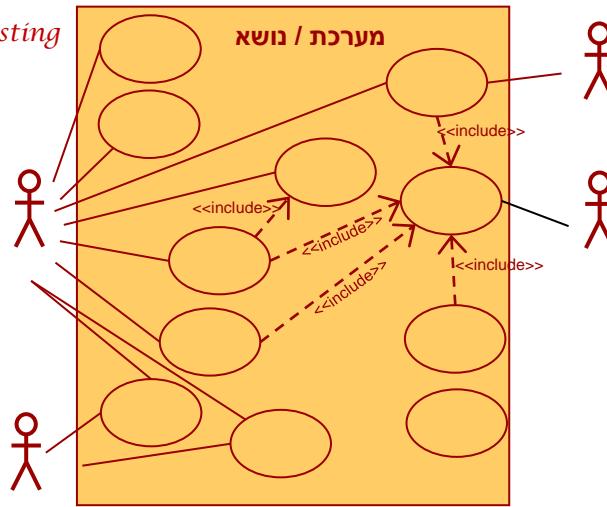
- על בסיס סיפור הלקוח ותרשים ה-UC בرمת הארגון
 - זהה תהליכי הממשקים את השירותים הארגוניים
 - תהליכי הממשיכים את השירותים הארגוניים
 - תהליכי הנדרשים לפעולה תקינה של המערכת
 - עורך Use Case Diagram למערכת

מפרט טקסטואלי של UC

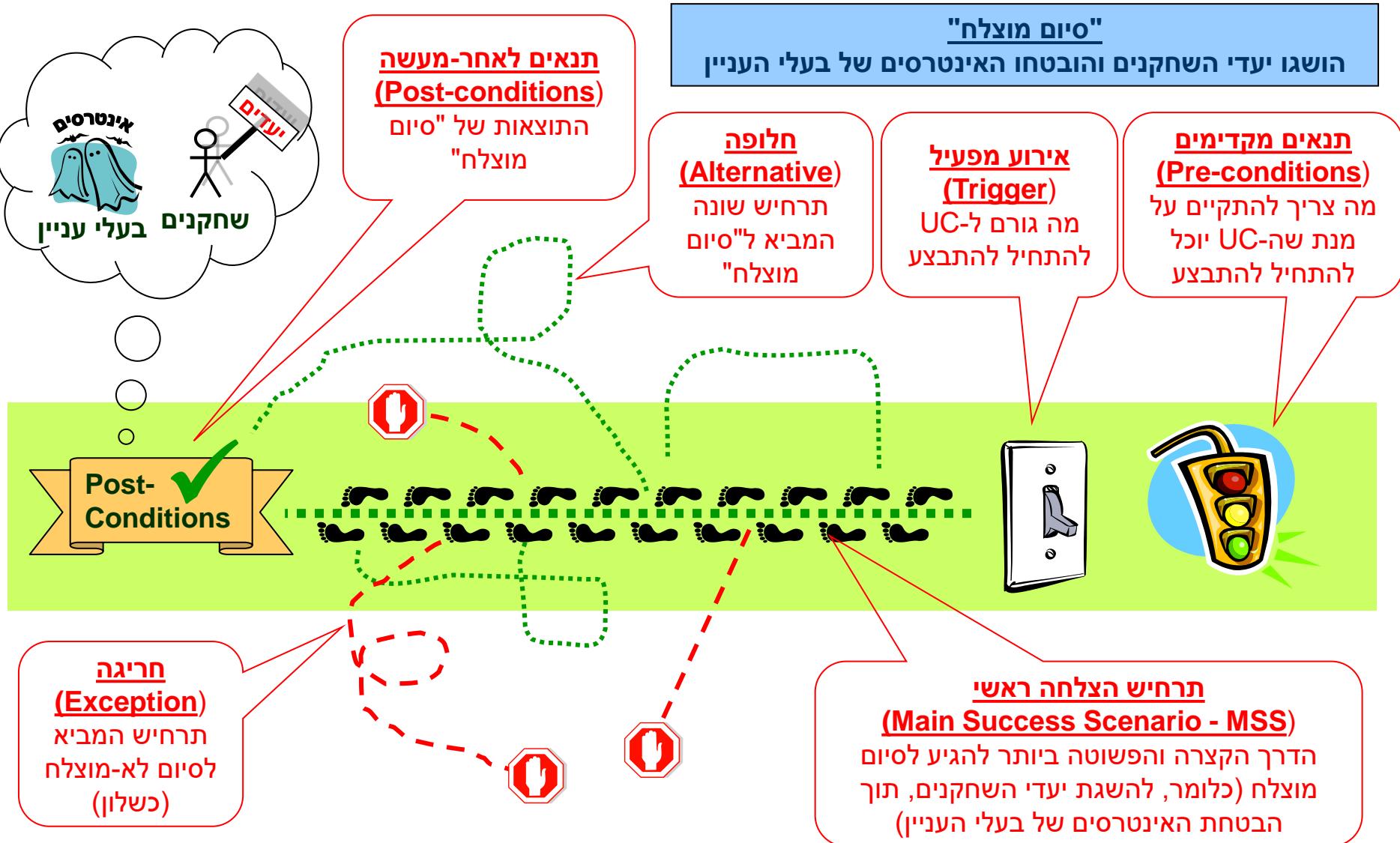


4. מפרט טקסטואלי

- **ניתוח דרישות**
- **Use Case Diagram**
- **השלבים:**
 1. **תיכון המערךת**
 2. **הגדרת שחקנים.**
 3. **תרשים UC**



מרכיבי Use Case ומಹלכו



קשרים מותניים בין Use Cases

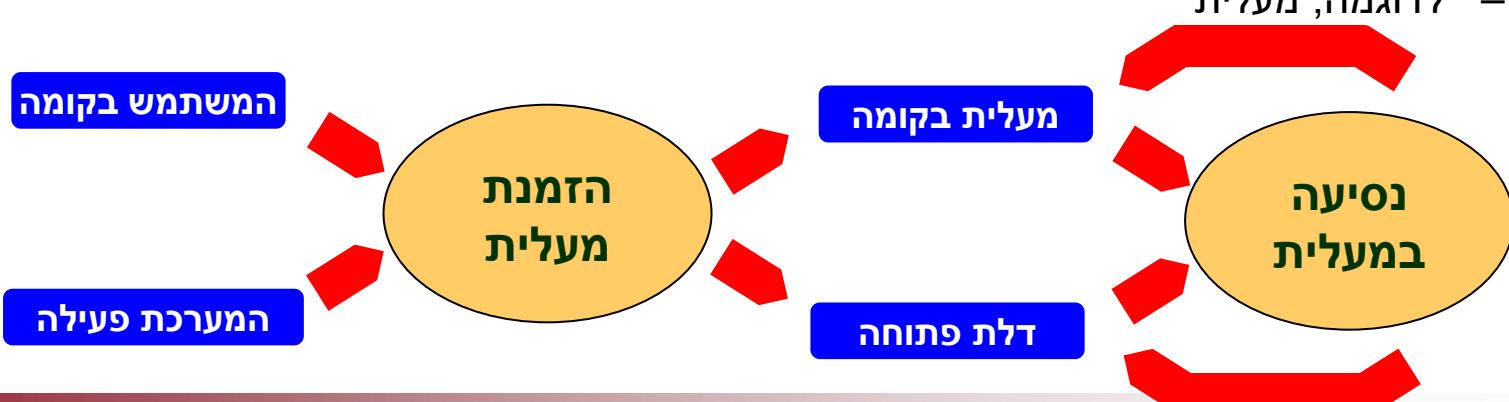
- Post-conditions של UC אחד יכולים להיות Pre-conditions של (s) UC אחר(ים)

– לדוגמה, כספומט



- זרימת הפעולות איננה בהכרח מוגדרת מראש (הפעלה היא נסיבתית – כאשר נוצרו התנאים ניתן לבצע)
- Pre-conditions יכולים להתקיים כתוצאה מהפעלת UC אחר

– לדוגמה, מעלית



מבנה לכתיבה מפורט טקסטואלי של Use Case

זהוי ה-UC	שם ה-UC [רצוי לבחור שם פעולה + מושא הפעולה, למשל: "ניהוג המטוס"]
שחקנים ויעדים	שחקן(ים) ראש(ים): מה יעדם בהפעלת UC זה, או "UC ספונטאני" שחקן(ים) תומך(ים): מה תפקידם ב-UC זה
בעלי עניין ואינטראסים	בעלי עניין שיש להם אינטראס ספציפי ב-UC זה, ופירוט האינטראס
Pre-conditions	תנאים והנחהות יסוד, שלא ניתן לא יכול ה-UC להתבצע. משפטים בוליאניים!
Post-Conditions	התוצאות של סיום מוצלח [מנקודות הראות של השחקן(ים) הראשי(ים)]. משפטים בוליאניים!
Trigger	האירוע (פנימי או יוזמת שחקן ראשי) הגורם ל-UC להתחיל לפעול
תרחיש הצלחה ראשי (MSS)	תיאור האינטראקציה ה"אידיאלית" בין השחקנים לבין "המערכת", מנקודת הראות של בעלי העניין החיצוניים , אשר תביא לקיום התנאים לאחר מעשה (Post-conditions): 1. המערכת <מגיבה ל-trigger> 2. שחקן כלשהו <עושה משהו> 3. המערכת <מגיבה באופן כלשהו> וכך הלאה, עד שימושים ה- post-conditions
הסתעפות #	חולפה/חריגה בצעד ... של תרחיש ...: <קרה שהוא אחר מהצפוי> סדרת צעדים חולפית המסתימה בהצלחה (חולפה) או בכשלון (חריגה)
עקיבות לדרישות	מספר דרישות (מטבליות הדרישות) הרלוונטיות ל-UC זה: דרישות תפעוליות המשפיעות על ביצוע ה-UC דרישות מידע הנוגעות לנתחנים רלוונטיים ב-UC דרישות לא-פונקציונאליות שיישפיעו על אופן המימוש של ה-UC בהמשך

UC הזמנת מעלית – מפרט טקסטואלי

הזמןת מעלית	UC-1
נוסף: לקבל מעלית זמינה לנסעה	শחקנים ויעדים
אין	ב"ע ואינטראסים
<ul style="list-style-type: none"> • הנוסף נמצא בקומת כלשהי בה נמצאת דלת של מעלית • המערכת פועלה (באמצעות UC "איתחול מערכת") 	Pre-conditions
<ul style="list-style-type: none"> • מעלית פתוחה נמצאת בקומת בה נמצא הנוסף (יעד) 	Post-Conditions
<ul style="list-style-type: none"> • הנוסף לוחץ על כפטור בקומת 	Trigger
<ol style="list-style-type: none"> 1. הcupטור שנלחץ נדלק 2. מעלית מגיעה לקומת 3. דלת המעלית נפתחת 4. כפטור הקומה כבוי 	MSS
אין	הסתעפויות
...	עקיבות לדרישות

UC נסעה במעלית – מפרט טקסטואלי (1)

נסעה במעלית	UC-2
נוסע: להגעה במעלית לקומת מボקשת	שחקנים ויעדים
מכבי- אש: נוסע לא יישאר תקוע במעלית	ב"ע ואינטראסים
<ul style="list-style-type: none"> • מעלית פתוחה נמצאת בקומת בה נמצא (בנסיבות UC "הזמןת מעלית") • המעלית פועלה (בנסיבות UC "איתחול מערכת") 	Pre-conditions
<ul style="list-style-type: none"> • הנוסע הגיע לקומת המבויקשת (יעד) • הנוסע יכול לצאת מהמעלית (אינטרס) 	Post-Conditions
<ul style="list-style-type: none"> • הנוסע נכנס למעלית 	Trigger
<ol style="list-style-type: none"> 1. הנוסע לוחץ על כפתור הקומה המבויקשת 2. הceptor שנלחץ נדלק 3. דלת המעלית נסגרת (במידה והיתה פתוחה) 4. המעלית ממשיכה בנסעה באותו כיוון, אם קיימות בקשות קודמות 5. המעלית נעצרת בקומת הבאה אליה קיימת בקשה 6. הדלת נפתחת 7. כפתור הקומה כבה 8. חוזרת לצעד 3, כל עוד לא הגיע הנוסע לקומת היעד 9. הנוסע יוצא מהמעלית 	MSS

המשר בשקף הבא...

UC נסיעה במעלית – מפרט טקסטואלי (2)

... המשך מהשיקף הקודם

<p>הסתעפות א'</p> <p>חלופה מצעד 4 של MSS: אין בקשות באותו כיוון / המעלית הגיעה לкраה המסלול</p> <p>4א. המעלית מחליפה את כיוון הנסעה</p> <p>4ב. התרחיש נמשר</p>	
<p>הסתעפות ב'</p> <p>חריגת בצעד 4 של MSS: הנושא לחץ על כפתור עצירת חירום</p> <p>4ב.1. המעלית נעצרת מיד</p> <p>4ב.2. המעלית מבטלת את כל בקשות העצירה הקיימות</p> <p>4ב.3. התרחיש מסתיים</p>	
<p>הסתעפות ג'</p> <p>חריגת בצעד 4 של MSS: המעלית נתקעה</p> <p>4ג.1. הנושא מזעיק חילוץ</p> <p>4ג.2. מתחיל חילוץ [extending UC-3]</p>	
<p>...</p>	<p>עקיבות לדרישות</p>

UC חילוץ נושא – מפרט טקסטואלי

חילוץ נושא	UC-3
נושא: להיחלץ ממעלית שנתקעה מחלץ: שחזור תונר	শחקנים ויעדים
שירותי ההצלה: נושא לא ישאר תקוע במעלית	ב"ע ואינטראסים
• המעלית תקועה (נתקעה במהלך נסיעה [2-UC])	Pre-conditions
• הנושא יכול לצאת מהמעלית (יעד + אינטרס)	Post-Conditions
• הנושא מזעיק חילוץ	Trigger
1. מחלץ מגיע לחדר המכונות 2. המחלץ מפעיל את המעלית ב-elevator mode חילוץ 3. המעלית מגיעה לקומת הקרקע 4. המחלץ פותח את המעלית 5. הנושא יוצא מהמעלית	MSS
אין	הסתעפויות
...	עקיבות לדרישות

UC בדיקת מערכת – מפרט טקסטואלי (1)

UC-4	בדיקות מערכת
שחקנים ויעדים	טכני: להשלים בדיקה מקיפה ול證ודא שהמערכת תקיןה
ב"ע ואינטראסים	נוסף+משע"ב: המערכת תקיןہ וב吐חה לשימוש (אמינות, זמינות, בטיחות)
Pre-conditions	<ul style="list-style-type: none"> • המערכת פועלה (באמצעות UC "איתחול מערכת") • המערכת לא נמצאת בשימוש
Post-Conditions	<ul style="list-style-type: none"> • כל פונקציות המערכת תקיןות • רשם דוח תקיןות מלא
Trigger	<ul style="list-style-type: none"> • הטכני מפעיל את תוכנת הבדיקה
MSS	<ol style="list-style-type: none"> 1. הטכני פותח דוח תקיןות חדש 2. הטכני מזמן מעלית לעליה בקומה שטרם נבדקה והمعالית מגיעה [UC-1] 3. הטכני מזמן מעלית לירידה באותו קומה והمعالית מגיעה [UC-1] 4. הטכני מסמן בדוח שהקומה תקיןہ 5. הטכני חוזר על צעדים 5-2 עבור כל הקומות 6. הטכני נכנס למעלית שטרם נבדקה 7. הטכני נושא בمعالית לכל הקומות [UC-2] 8. הטכני מסמן בדוח שהمعالית תקיןہ 9. הטכני חוזר על צעדים 9-7 עבור כל המעליות

UC בדיקת מערכת – מפרט טקסטואלי (2)

... המשך מהשוף הקודם

<p>חולפה מצעד S ($S = 2, 3$ או 7) של MSS: תקלת בקומה/מעלית כלשהי</p> <p>Sא.1. הטכני מתקן את התקלה [UC-5]</p> <p>Sא.2. הטכני חוזר על הבדיקה שנכשלה ומבודד שהתקלה תוקנה</p> <p>Sא.3. התרחיש נמשר</p>	הסתעפות א'
<p>חריגה בצעד Sא.1 של הסתעפות א': הטכני לא הצליח לתקן את התקלה</p> <p>Sא.1ב.1. הטכני מסמן את התקלה בדו"ח</p> <p>Sא.1ב.2. עברו לצעד 5 או 9, בהתאם</p>	הסתעפות ב'
...	עקיבות לדרישות

UC איתחול מערכת – מפרט טקסטואלי

UC-6	איתחול מערכת
שחקנים ויעדים	איש האחזקה: הכנסת המערכת המעליות לפעולה
ב"ע ואינטראסים	נוסעים: המערכת זמינה לשימוש
Pre-conditions	<ul style="list-style-type: none"> המערכת איננה פועלת
Post-Conditions	<ul style="list-style-type: none"> המערכת פועלה ומאפשרת שימוש
Trigger	<ul style="list-style-type: none"> איש האחזקה מודיעק את המערכת
MSS	<ol style="list-style-type: none"> המערכת מתחילה את כל רכיביה (מעליות, פאנלים וכו') המערכת מודדת שכל מרכיביה תקין המערכת מציגה חיוי תקינות
הסתעפויות	<p>חריגת בצעד 1 של MSS: לא כל מרכיבי המערכת תקין</p> <p>1א. המערכת מציגה חיוי אי-תקינות התרחיש מסתים</p>
עקיפות לדרישות	...

עקיבות הדרישות התפעוליות למודל UC

UC	סוג	נוטח	זיהוי
		▪▪▪▪▪	
UC-1	OR	נוסע הנמצא בקומה כלשהי ... לוחץ על הכפתור המתאים לכיוון הנסיעה המבוקש	
UC-1	OR	כפתור קומה נדלק בעקבות לחיצה, אם לא היה דלוק קודם לכן	
UC-1	OR	בעקבות לחיצה על כפתור בקומה, מעלית כלשהי הנמצאת בכיוון הנסיעה המבוקש תגיע לקומת	
UC-1	OR	עם הגעת מעלית לקומת תיפתח הדלת וכפתור הקומה יכבה	
		▪▪▪▪▪	
UC-2	OR	נוסע הנמצא בתוך המעלית ורוצה להגיע לקומת כלשהי לוחץ על הכפתור המתאים	
UC-2	OR	כפתור מעלית נדלק בעקבות לחיצה, אם לא היה דלוק קודם	
UC-2	OR	בעקבות לחיצה על כפתור במעלית למעלה תתווסף בקשה לעצירה בקומה המתאימה	
UC-2	OR	בעקבות הגעת מעלית לקומת מבקשת היא תיעצר, הדלת תפתח והכפתור המתאים לקומת יכבה	
UC-3	OR	איש האחזקה ... הוכשר לחילוץ נוסעים תקועים ...	
UC-4	OR	... המערכת נבדקת... בידי טכנאי מוסמך	
UC-5	OR	אם הטכנאי מגלה תקלת הוא מנסה לתקן אותה...	

- על בסיס סיפור הלקוח ותרשים ה-UC של המערכת
 - כתוב מפרט טקסטואלי עבור אחד מה-UC המשמעותיים
 - בטלת הדרישות (גרסה 1) זהה את הדרישות הפעוליות המקבלות מענה ב-UC זה.

Use Cases ודרישות תפעוליות

- **מהווים את הבסיס לימוש ההתנהגות הכללת של המרבית, לכן**
 - כל Use Case צריך להתייחס לדרישה תפעולית אחת לפחות
 - **אחרת הוא מיותר**
 - כל דרישת פעולה חייבת להיות מוקנית ל-Use Case אחד לפחות
 - **אחרת היא לא תמומש במערכת**
- **מודל Use Cases השלם של מערכת-העניין חייב לכנות את כל הדרישות התפעוליות**
- **עקיבות דו-כיוונית בין המודל לבין מאגר הדרישות היא חיונית!**

Use Cases ודרישות לא-פונקציונאליות

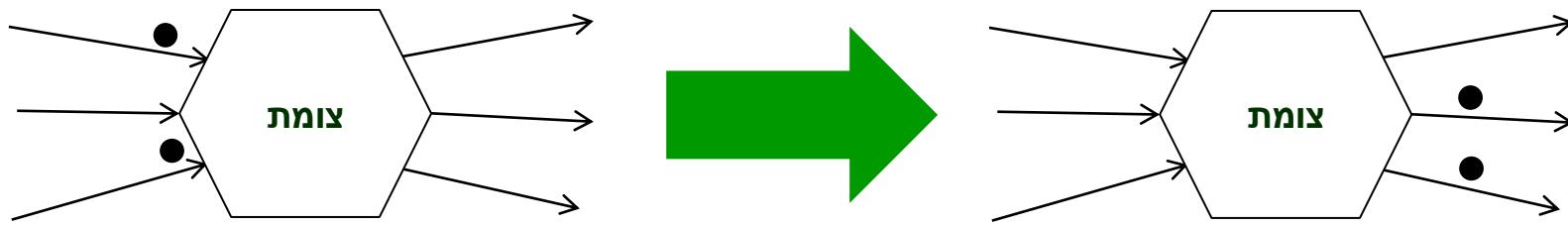
- דרישות לא פונקציונאליות (מאפייני איקות) מגדירים "כמה טוב" לבצע המערכת את הפעולות שלה
 - ביצועים
 - אמינות / זמינות
 - בטיחות / אבטחת מידע
 - מדי אפקטיביות (MoE)
- מאפייני האיקות מושגים במערכת במהלך הארכיטקטורה והתוכן
 - בחירת ארכיטקטורה/תוכן שיספקו את המאפיינים הנדרשים
- לפיכך, מודל ה-UC, מטבעו, אינו יכול, בדרך כלל, להתייחס לדרישות לא-פונקציונאליות
- יחד עם זאת, על המודל להציג על דרישות לא-פונקציונאליות רלוונטיות שייהי צורך לקחת בחשבון בזמן יימוש ה-UC
- יצא דופן הוא מאפיין **השימושיות** העשי להיות מושפע באופן משמעותי מAFX האינטראקציה של המערכת עם משתמשה, הנקבעת בתרחישי h-UC Use Cases

ניתוח והגדרה מפורטת של תהליכיים

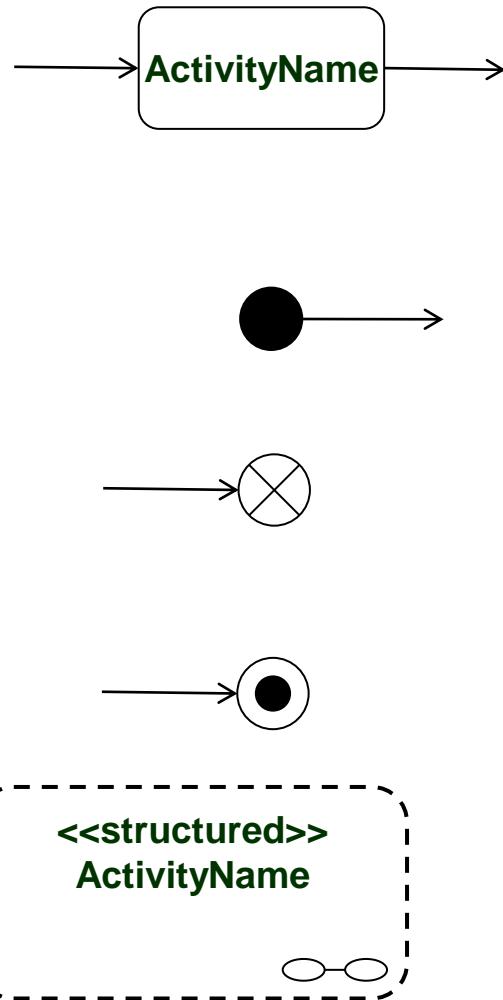
- מודל Use Case הגידיר את התהליכיים מנקודת הראות החיצונית
 - האינטראקציה של השחקנים עם המערכת
- על מנת להמשיך בתהליך התכנן יש צורך לתאר את התהליכיים כפי שעלייהם לבוא לידי ביטוי בפעולת המערכת
 - בשלב זה עדין לא פירקנו את מערכת-הענין למרכיביה, ולכן התהליך יתיחס למערכת הענין כאל מרכיב אלמנטרי, ללא פירוק פנימי
 - **תיאור תהליכי המערכת = לוגיקת הפעולה של המערכת**
- נכיר שני מודלים לתיאור התנהבות של מערכת כרכיב אלמנטרי
 - מודל פעילות – Activity Model
 - **יוזג כתעת**
 - מודל מכונת מצבים – State Machine Model
 - **יוזג בהמשך**

תרשים פעילות – Activity Diagram

- תרשים המשמש לתיאור זרימה (flow) של תהליכיים
- מורכב מצמתים (nodes) ומחיצים חד-כיווניים המחברים ביניהם
- מנגנון הפעולה: העברת אסימונים (tokens)
 - האסימונים נמצאים על החיצים
 - כל צומת משמש כצרך אסימונים וכייצן אסימונים
 - לכל סוג של צומת יש כללי העברת אסימונים
 - צומת יכול "לפעול" כאשר יש לו מספיק אסימונים בכניסה, על פי הכלל המתאים
 - עם תחילת הפעולה צורך (consumes) הצומת את כל האסימונים שבכניסה
 - עם סיום הפעולה מייצר (produces) הצומת אסימונים חדשים ביציאותיו, על פי הכלל המתאים



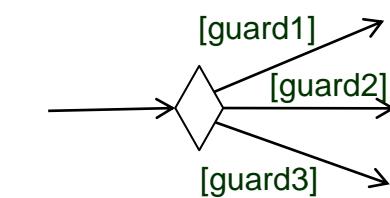
צמתי פעילות ופעולות – Activity Diagram



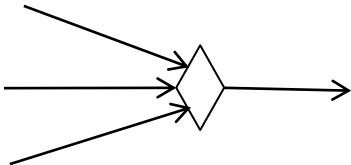
- **פעילות/פעולה (activity/action)**
 - כניסה אחת ויציאה אחת
 - הצומת יכול לפעול כאשר יש לו אסימון בכניסה
 - עם תחילת הפעולה נדרש הצומת את האסימון שבכניסה
 - עם סיום הפעולה מפיק הצומת אסימון ביציאה
- **התחלת פעילות (ActivityInitial)**
 - יציאה אחת ללא כניסה
 - עם הפעלה מפיק הצומת אסימון אחד ביציאה
- **סיום זרימה (FlowFinal)**
 - כניסה אחת, ללא יציאות
 - עם הפעלה נדרש הצומת את האסימון שבכניסה
- **סיום פעילות (ActivityFinal)**
 - כניסה אחת, ללא יציאות
 - עם הפעלה נעצרת כל פעילות התרשים
- **פעילות מובנית (structured activity)**
 - פעולה-על המכילה תרשימים פעילות

• **סוגים שונים: <>loop , <>sequential , <>conditional**

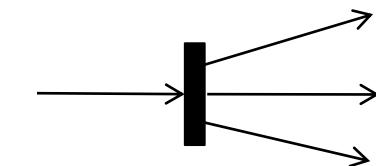
הנדסת תוכנה – צמתים בקרה – Activity Diagram



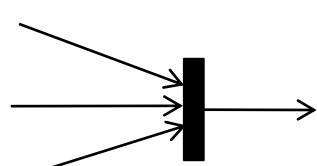
- החלטה (decision)
 - כניסה אחת, יציאות רבות
 - הצומת יכול לפעול כאשר יש לו אסימון בכניסה
 - פעולה הצומת מצרכת את אסימון הכניסה ומפיקה אסימון באחת היציאות בלבד, על פי התנאי [guard] שמתקיים (אופרטור לוגי XOR)



- התמזגות (merge)
 - יצאה אחת, כניסה רבות
 - הצומת יכול לפעול כאשר יש לו אסימון בכניסה אחת לפחות (אופרטור לוגי OR)
 - פעולה הצומת מצרכת את כל אסימוני הכניסה ומפיקה אסימון אחד ביציאה

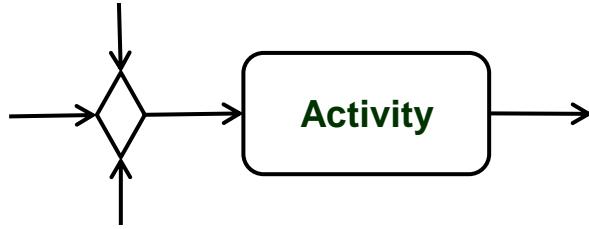


- מזלג (fork)
 - כניסה אחת, יציאות רבות
 - הצומת יכול לפעול כאשר יש לו אסימון בכניסה
 - פעולה הצומת מצרכת את אסימון הכניסה ומפיקה אסימונים בכל היציאות (אופרטור לוגי AND)
 - מצין שכל הפעולות היוצאות ממנה מתחילהות יחד.

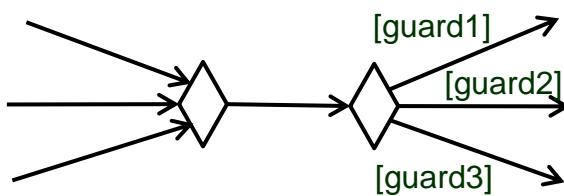
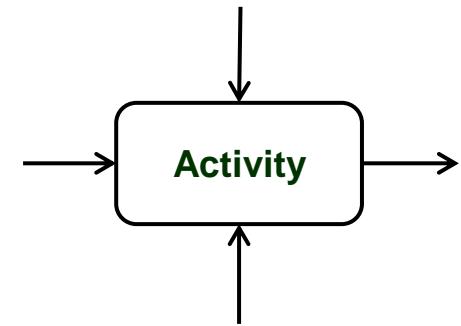


- הצלפות (join)
 - כניסה רבות, יצאה אחת
 - הצומת יכול לפעול אך ורק כאשר יש אסימונים בכל הכניסות (אופרטור לוגי AND)
 - פעולה הצומת מצרכת את כל אסימוני הכניסה ומפיקה אסימון אחד ביציאה
 - מצין שכל הפעולות הנכנסות אליו מסתיימות יחד.

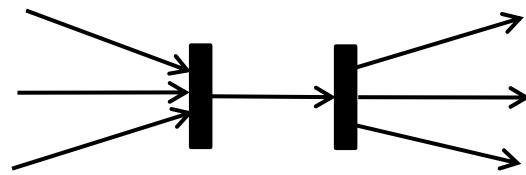
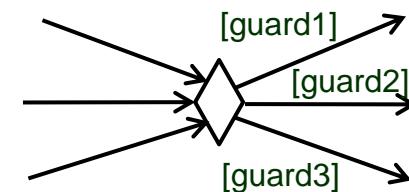
קיצורים תחביריים



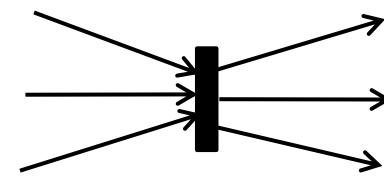
שקל ל...



שקל ל...



שקל ל...



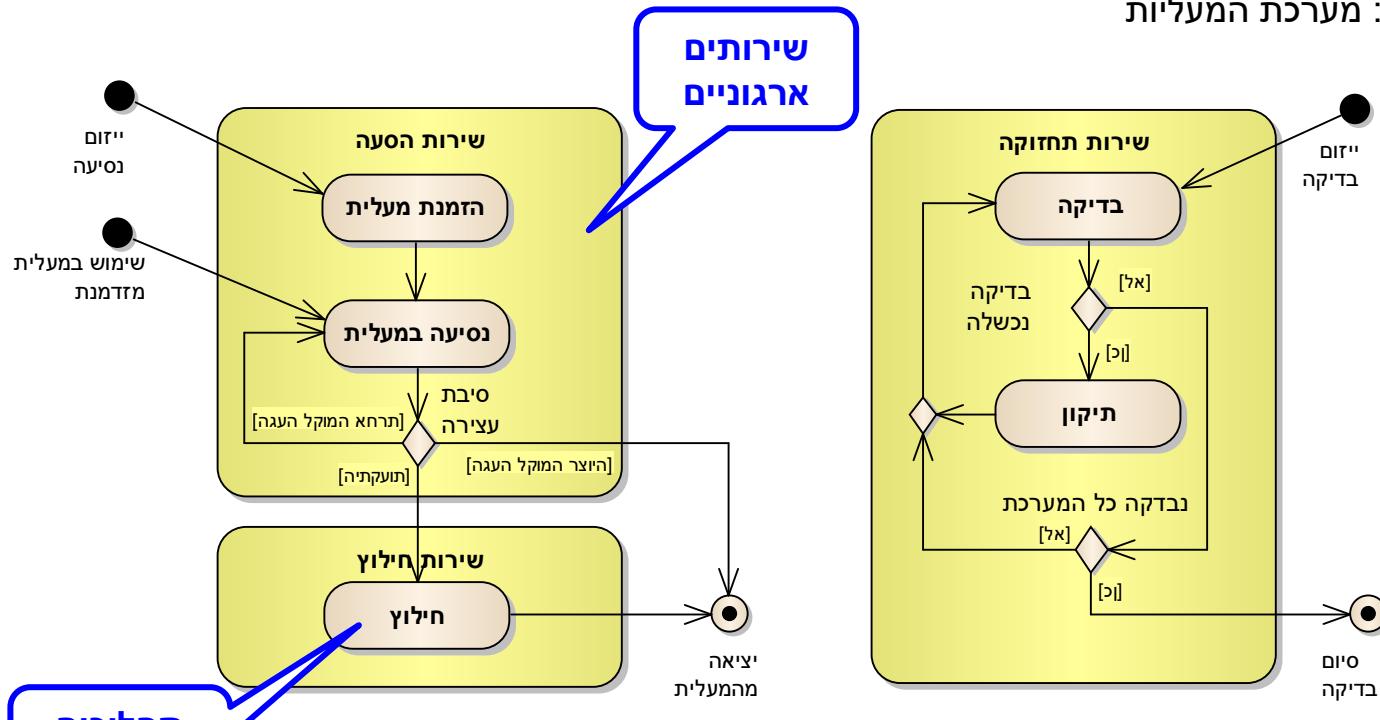
מערכת המעליות – לוגיקת התפעול המערכתית

- לוגיקת התפעול המערכתית

- האופן בו מפעילים תהליכי מערכת לצורך קבלת השירותים הארגוניים
 - התהליכים מועמדים להיות Use Cases בرمת המערכת

- ניתן למדל את לוגיקת התפעול באמצעות תרשימים פעילות (Activity Diagram)

- לדוגמה: מערכת המעליות



- נחזר לתרשיימי פעילות בהמשך

- כרגע נתקדם עם התהליכים שziehainו אינטואיטיבית

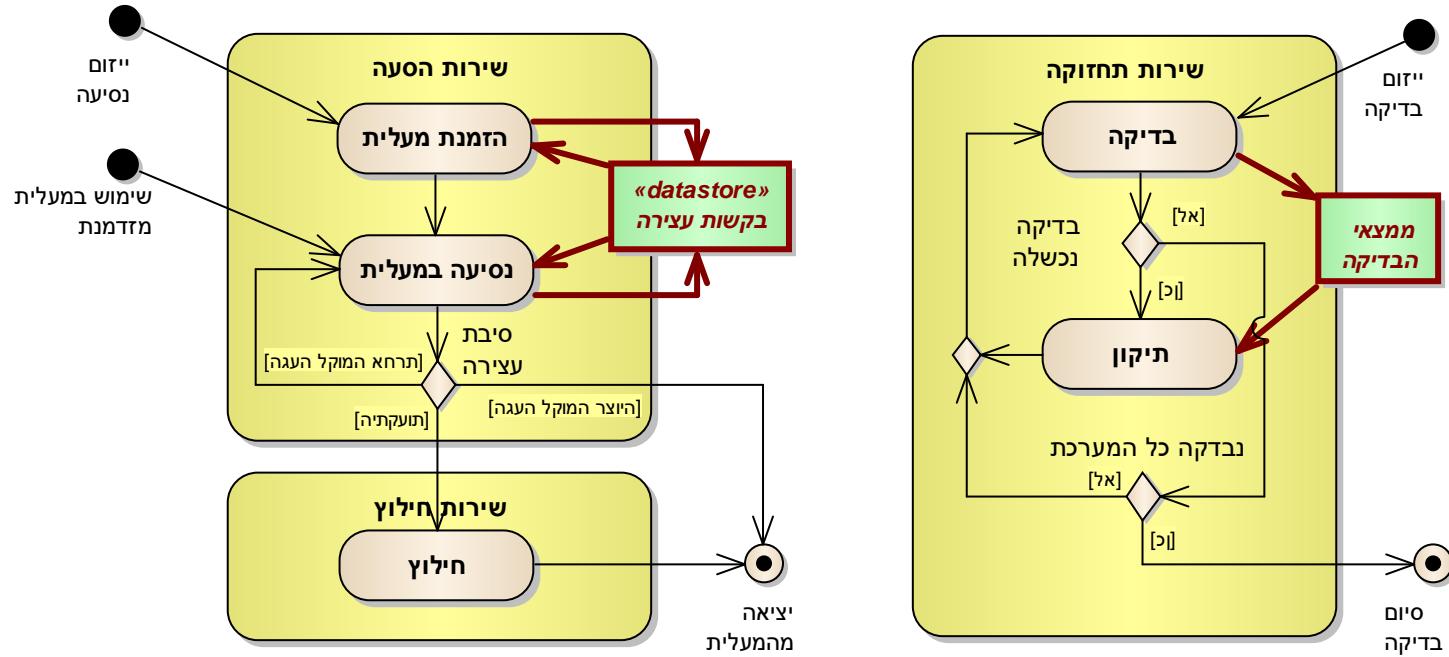
זרימת נתונים (data flow) בתרשימים פעילים

- בנוסף לזרימת הבקירה, ניתן ליצג בתרשימים פעילים גם זרימת נתונים בין הפעולות השונות
- אובייקט (Object)
 - פריט מידע בודד, המהווה קלט/פלט של פעולה
- מאגר נתונים (Data Store)
 - פעולות יכולות לאחסן בו או לשולוף ממנו פריטי מידע
- צמתי הבקירה וכללי פעולתם חלים גם לגבי זרימת מידע



מערכת המעליות: לוגיקת פעולה + זרימת נתונים

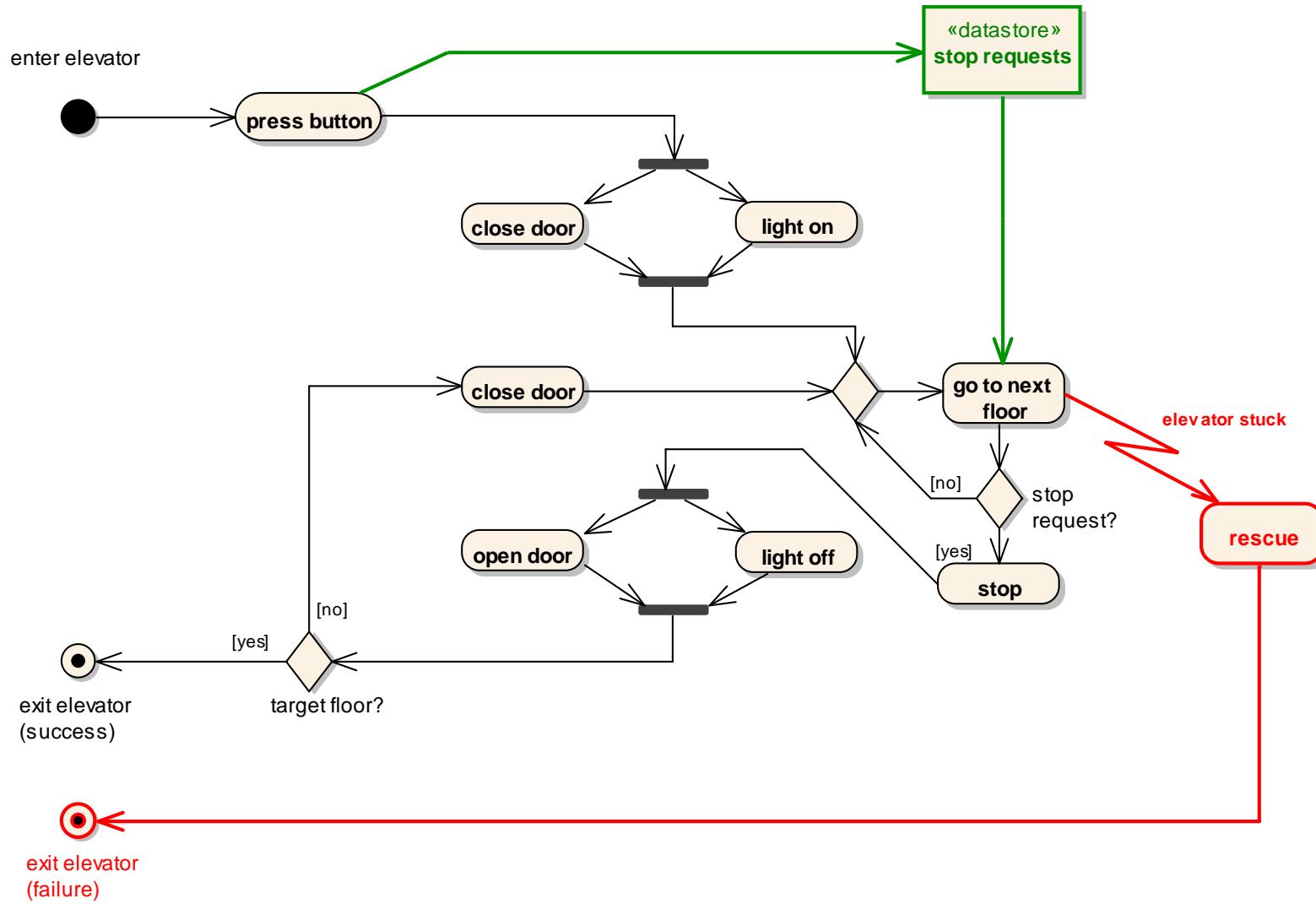
control flow + data flow



פירוט תרחishi UC באמצעות Activity Diagram

- ניתן להשתמש בתרשימים פועלות (Activity Diagram) לתיאור התרחישים של case use באופן גרפי
 - נקודת המבט משתנה:
 - מפרט טקסטואלי – מנקודת ראות המשתתפים (אינטראקטיה)
 - (process flow – מנקודת ראות המערכת (Activity Diagram
 - ניתן לכלול בתרשימים אחד הן את תרחיש הצלחה הראשי והן את הרשותפויות
 - ניתן לבטא בתרשימים את "חלוקת העבודה" בין המשתתפים באמצעות סימון "מסלולי שחיה" (swim lanes)
 - **לתשומת לב:** התרשימים מחליף אך ורק את האירוע המפעיל + תיאור התרחישים
 - את כל שאר חלקיו ה-UC (post-cond, pre-cond וכו') יש לפרט בטקסט!

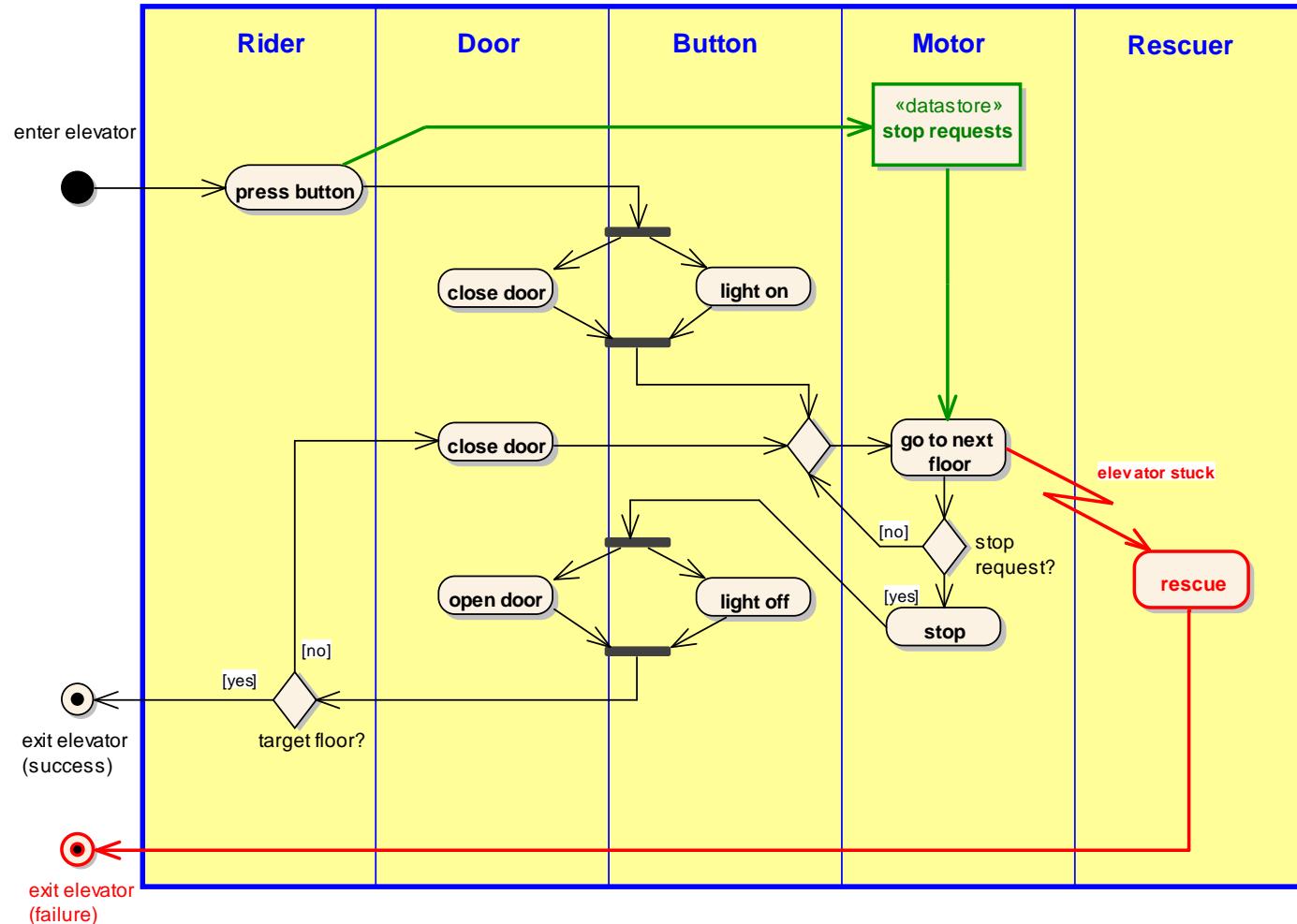
Activity Diagram – UC-2 נסיעה במעלית





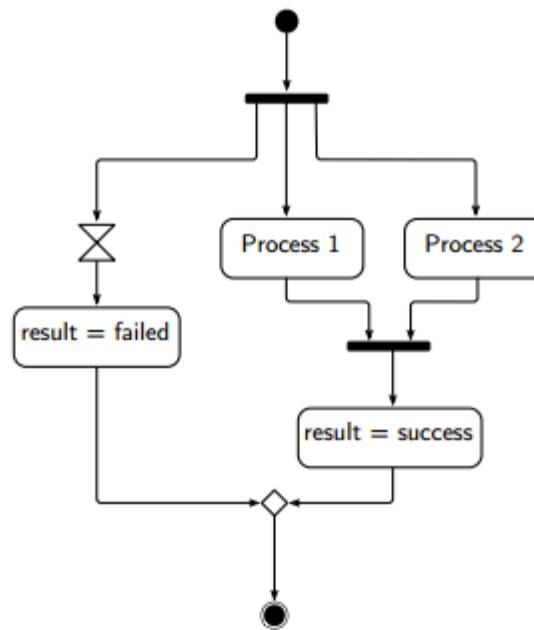
מסלולי שחיה (Swim Lanes) (Swim Lanes)

- סימון מחיצות בתרשים, המציגות את חלקו של כל אחד מהמרכיבים בפעולות המערכתית המשותפת.



שימוש ב-**Timer**

דוגמא: לאחר איתחול מערכת, Timer ושני תהליכי מאוחלים. המערכת שומרת את התוצאה וערכה
במידה ושני התהליכים הסתיימו לפני פקיעת ה-Timer.
הסימון



- על בסיס המפרט הטקסטואלי של ה-UC מהמטלה הקודמת
 - זזה את היחסות המשתתפות ב-UC והגדיר להן מסלולי שחיה
 - ערכו ערך Activity Diagram (זרימת בקרה בלבד) שישקף את כל תרחישי UC זה
 - הוסף זרימת מידע לתרשים