פרופ' עמיר תומר
amir@amirtomer.com

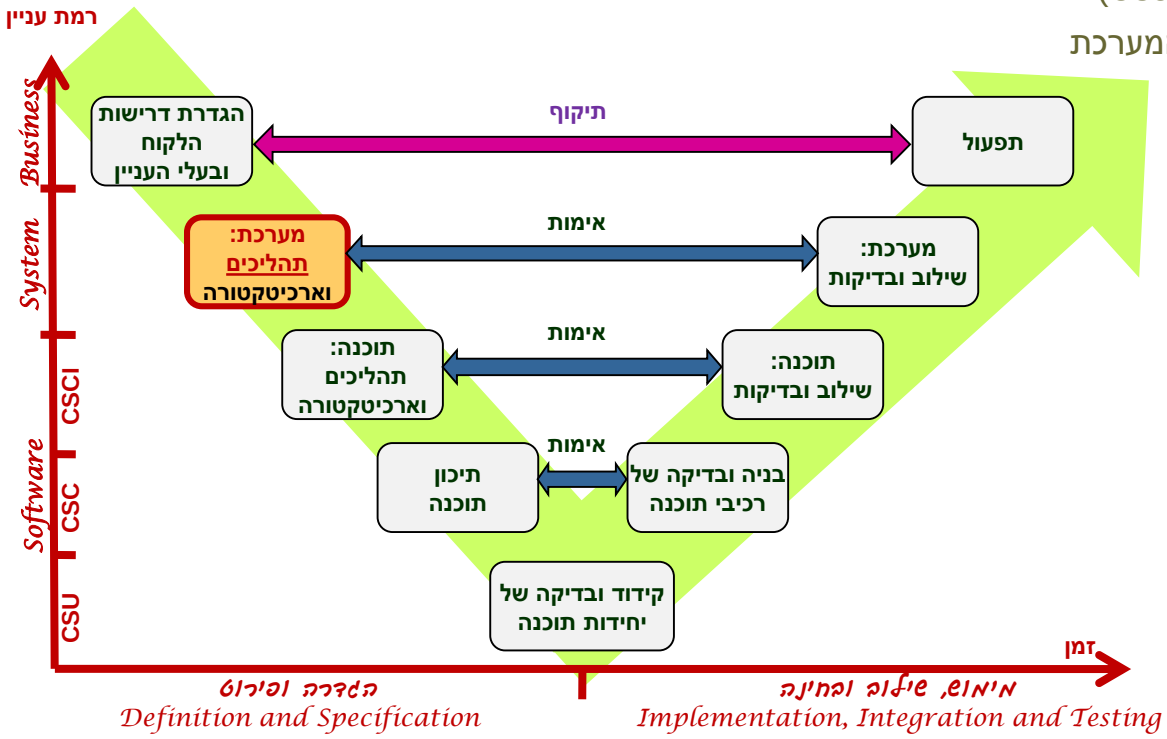
פרק 4

ניתוח דרישות והגדרת תהליכי מערכת

Requirements Analysis and Use Case Specification

פעילות ניתוח הדרישות והגדרת תהליכי המערכת

- מטרת הפעילות
  - הפיכת אוסף הדרישות וסיפורי הלקוח לתהליכים סדורים ומובָנים
- קלט
  - טבלת הדרישות, סיפור הלקוח ובעלי העניין
- תוצרים
  - תהליכי הארגון (הלוגיקה העסקית)
  - מפרט תהליכי מערכת (Use Cases)
  - עקיבות בין הדרישות לתהליכי המערכת



מתוך הערך waze בויקיפדיה

waze היא אפליקציית ניווט GPS חינמית ושיתופית לטלפונים ניידים חכמים

השימוש העיקרי ב-waze הוא באמצעות חיבור מקוון לאינטרנט שבאמצעותו המפות והמידע נטענים, אם כי קיימת אפשרות להשתמש בתוכנה ללא חיבור, כאשר המפות נטענו קודם לכן (הניווט אינו יעיל במצב זה).

הניווט באמצעות waze משרת שתי מטרות עיקריות:

הגעה ליעד שהמשתמש יודע את כתובתו אך אינו מכיר את הדרך אליו.

הגעה ליעד שהמשתמש מכיר את הדרך אליו, תוך בחירת חלופת הדרך המהירה ביותר (לא בהכרח הקצרה ביותר) להגעה ליעד זה, בהתחשב בעומסי התנועה בחלופות השונות.

התוכנה פועלת בצורה שיתופית והיא מתעדכנת על ידי קהילת המשתמשים. המידע המתקבל מכל משתמש, מועבר לשאר המשתמשים בתוכנה כדי לעדכן כבישים, מקומות ומספרי בתים, כמו גם כדי להתריע על עומסי תנועה, ניידות משטרה ומכשולים בכביש. בנוסף, נשלח אנונימית מכל משתמש מידע הכולל מהירות ומיקום, ומשמש את האפליקציה לניווט יעיל.

- האם תיאור זה יכול לשמש כבסיס לבניית המערכת?
- מה חסר בו?

תרחישי פעולה

- הדרישות, כשלעצמן, מפרטות את **התכונות והיכולות** שיש להקנות למערכת על מנת להשיג את מטרותיה
  - במערכת עתירת תוכנה התכונות והיכולות באות לידי ביטוי תוך כדי ביצוע **תרחישי הפעלה**
    - באינטראקציה עם הסביבה החיצונית
    - בתהליכים פנימיים (לוגיקת זרימה)
    - באינטראקציה בין המרכיבים הפנימיים
- האפיון התפעולי ("סיפור הלקוח"), אם קיים, מתאר את האופן בו המערכת פועלת
  - הגדרת נסיעה חדשה (הגדרת יעד, חישוב מסלולים, בחירת מסלול אופטימלי, ...)
  - ניווט לאורך מסלול נתון (איכון מיקום הרכב, הצגת המפה והמסלול, זיהוי סטיה וחישוב מסלול מחדש, ...)
  - דיווח על מפגעים (איכון המפגע, רישום פרטים, שליחה לשרת, ...)

לוגיקה עסקית (Business Logic)

- לוגיקה עסקית מתארת את תרחישי התפעול העקרוניים, באמצעותם ניתן לקבל את שירותי העסק/הארגון, תוך שימוש במערכת (מבלי להיכנס לפרטי המימוש)
  - שימוש במערכת (דו-שיח עם משתמש)

    ש: כיצד משתמשים במערכת?

    ת: נכנסים למעלית הנמצאת בקומת המוצא ונוסעים בה עד לקומת היעד

    ש: ואם אין מעלית זמינה בקומת המוצא?

    ת: אפשר להזמין מעלית, ואחת מהן תגיע

    ש: ומה קורה אם המעלית נתקעת תוך כדי נסיעה?

    ת: לוחצים על כפתור האזעקה ומחלץ יגיע.
  - תחזוקת מערכת (דו שיח עם טכנאי)

    ש: כיצד מבצעים תחזוקה למערכת?

    ת: מבצעים בדיקה כוללת למערכת ומוודאים שהיא תקינה

    ש: ואם התגלתה תקלה?

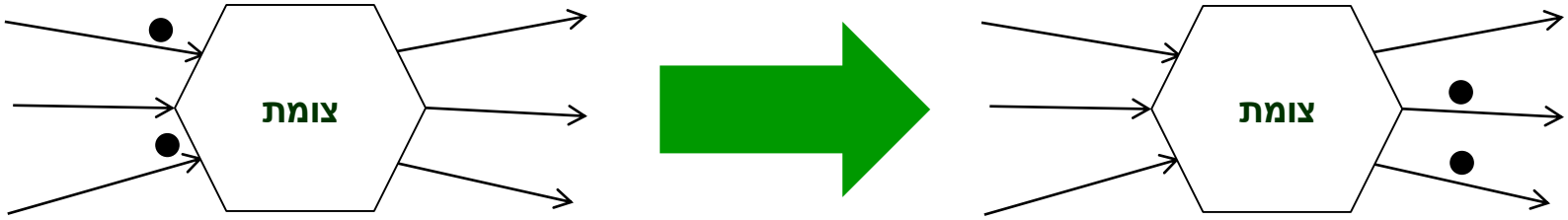
    ת: אם היא ניתנת לתיקון במקום – מתקנים אותה וחוזרים על הבדיקה

    ש: ואם לא?

    ת: חוזרים בפעם אחרת, מתקנים ובודקים שוב.

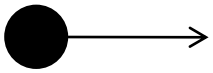
תרשים פעילות – Activity Diagram

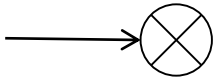
- תרשים UML המשמש לתיאור זרימה (flow) של תהליכים
- מורכב מצמתים (nodes) ומחיצים חד-כיווניים המחברים ביניהם
- <u>מנגנון הפעולה</u>: העברת אסימונים (tokens)
  - האסימונים נמצאים על החיצים
  - כל צומת משמש כצרכן אסימונים וכיצרן אסימונים
  - לכל סוג של צומת יש כללי העברת אסימונים
  - צומת יכול "לפעול" כאשר יש לו מספיק אסימונים בכניסה, על פי הכלל המתאים
  - עם תחילת הפעולה צורך (consumes) הצומת אסימונים מהכניסה
  - עם סיום הפעולה מייצר (produces) הצומת אסימונים חדשים ביציאותיו, על פי הכלל המתאים

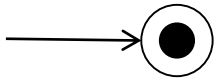


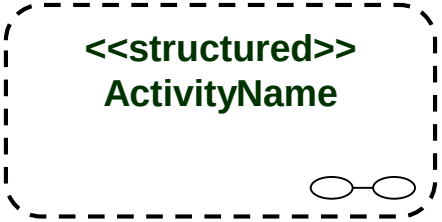
Activity Diagram – צמתי פעילויות ופעולות

- פעילות/פעולה (activity/action)
  - כעיקרון, כניסה אחת ויציאה אחת
  - הצומת יכול לפעול כאשר יש לו אסימון בכניסה
  - עם תחילת הפעולה צורך הצומת את האסימון שבכניסה
  - עם סיום הפעולה מפיק הצומת אסימון ביציאה
- התחלת פעילות (ActivityInitial)
  - יציאה אחת ללא כניסות
  - עם הפעלתו מפיק הצומת אסימון אחד ביציאה
- סיום זרימה (FlowFinal)
  - כניסה אחת, ללא יציאות
  - עם הפעלתו צורך הצומת את האסימון שבכניסה
- סיום פעילות (ActivityFinal)
  - כניסה אחת, ללא יציאות
  - עם הפעלתו נעצרת כל פעילות התרשים
- פעילות מובנית (structured activity)
  - פעילות-על המכילה תרשים פעילות
    - סוגים שונים: <<loop>>, <<sequential>> , <<conditional>>



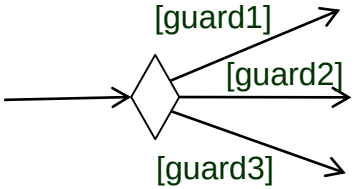




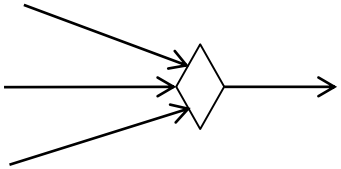


Activity Diagram – צמתי בקרה

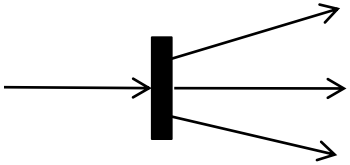
- **החלטה (decision)**
  - כניסה אחת, יציאות מרובות
  - הצומת יכול לפעול כאשר יש לו אסימון בכניסה
  - פעולת הצומת צורכת את אסימון הכניסה ומפיקה אסימון <u>באחת היציאות בלבד</u>, על פי התנאי [guard] שמתקיים (אופרטור לוגי XOR)



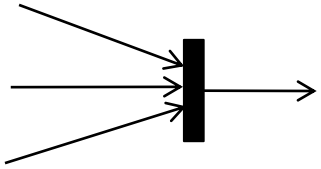
- **התמזגות (merge)**
  - יציאה אחת, כניסות מרובות
  - הצומת יכול לפעול כאשר יש לו אסימון <u>בכניסה אחת</u> לפחות (אופרטור לוגי OR)
  - פעולת הצומת צורכת <u>אחד מאסימוני הכניסה</u> ומפיקה אסימון אחד ביציאה



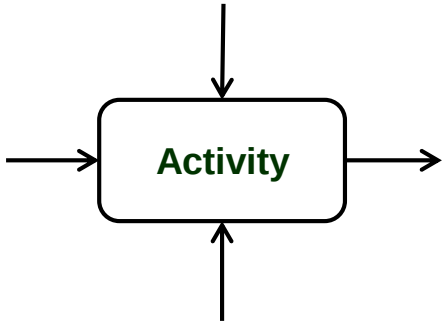
- **מזלג (fork)**
  - כניסה אחת, יציאות מרובות
  - הצומת יכול לפעול כאשר יש לו אסימון בכניסה
  - פעולת הצומת צורכת את אסימון הכניסה ומפיקה אסימונים <u>בכל היציאות</u> (אופרטור לוגי AND)



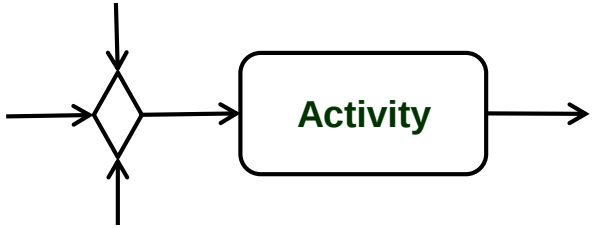
- **הצטרפות (join)**
  - כניסות מרובות, יציאה אחת
  - הצומת יכול לפעול אך ורק כאשר יש אסימונים <u>בכל הכניסות</u> (אופרטור לוגי AND)
  - פעולת הצומת צורכת את <u>כל אסימוני הכניסה</u> ומפיקה אסימון אחד ביציאה

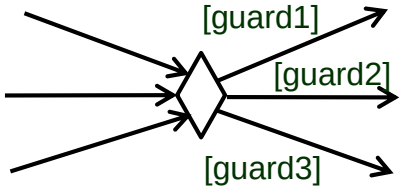


קיצורים תחביריים

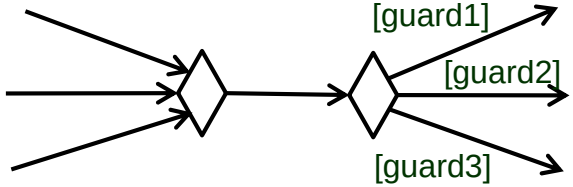


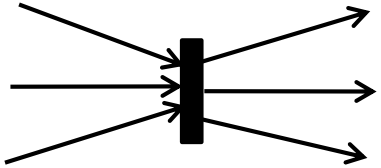
שקול ל...



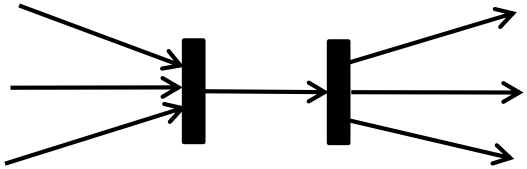


שקול ל...



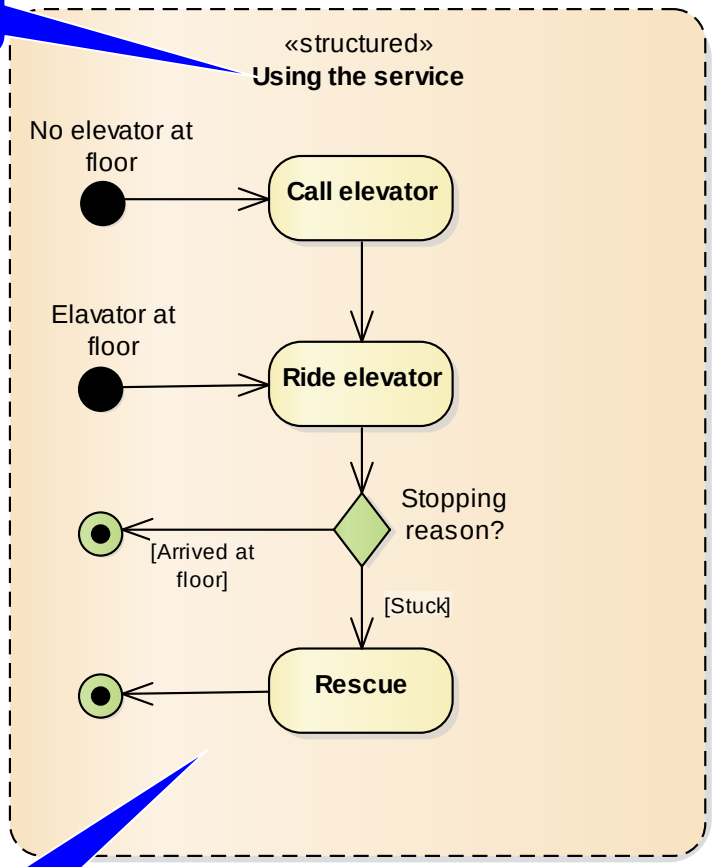


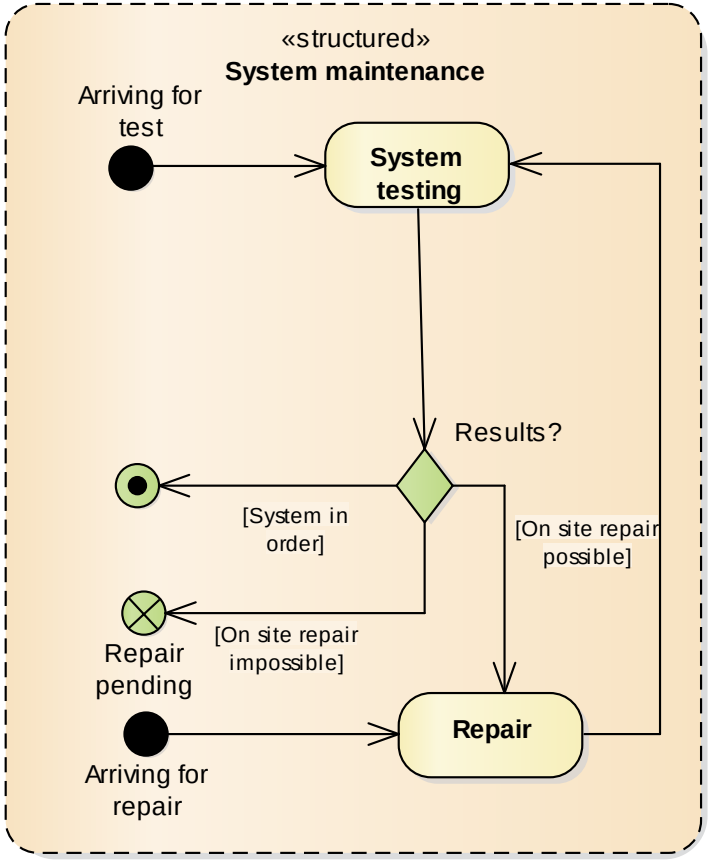
שקול ל...



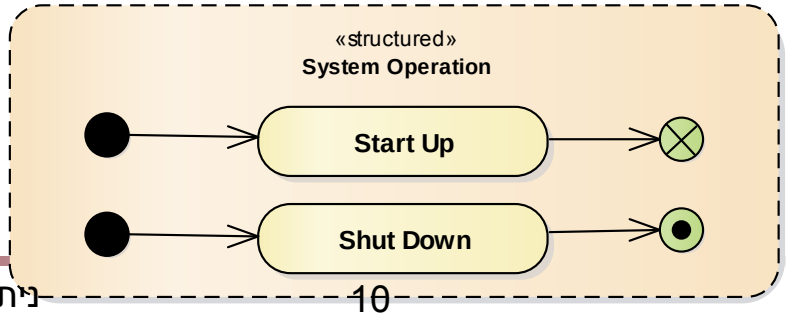
תיאור הלוגיקה העסקית של שירות המעליות באמצעות Activity Diagram

שירות של הארגון





תפעול השירות תוך שימוש במערכת



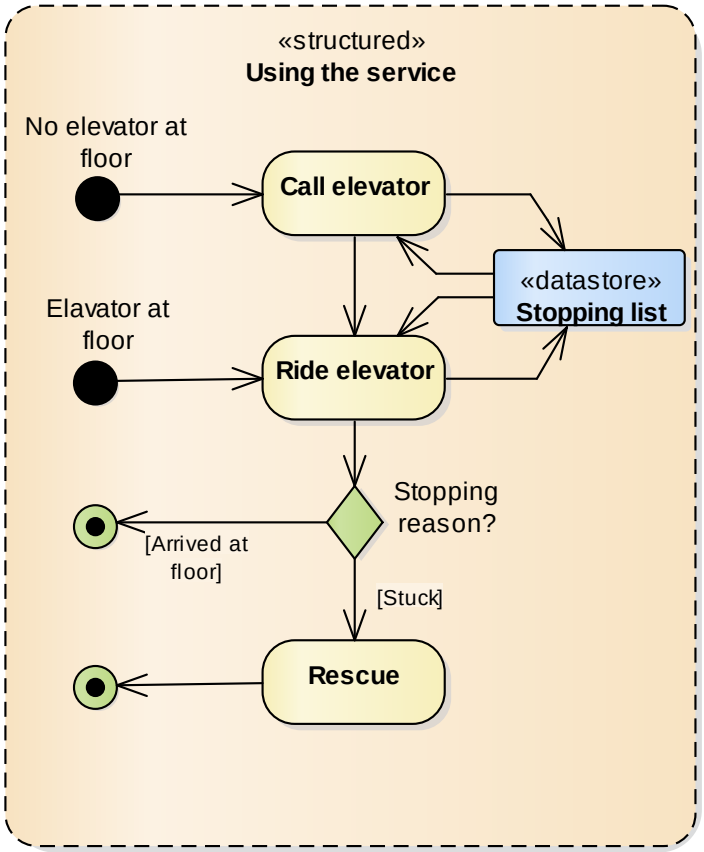
זרימת נתונים (data flow) בתרשים פעילות

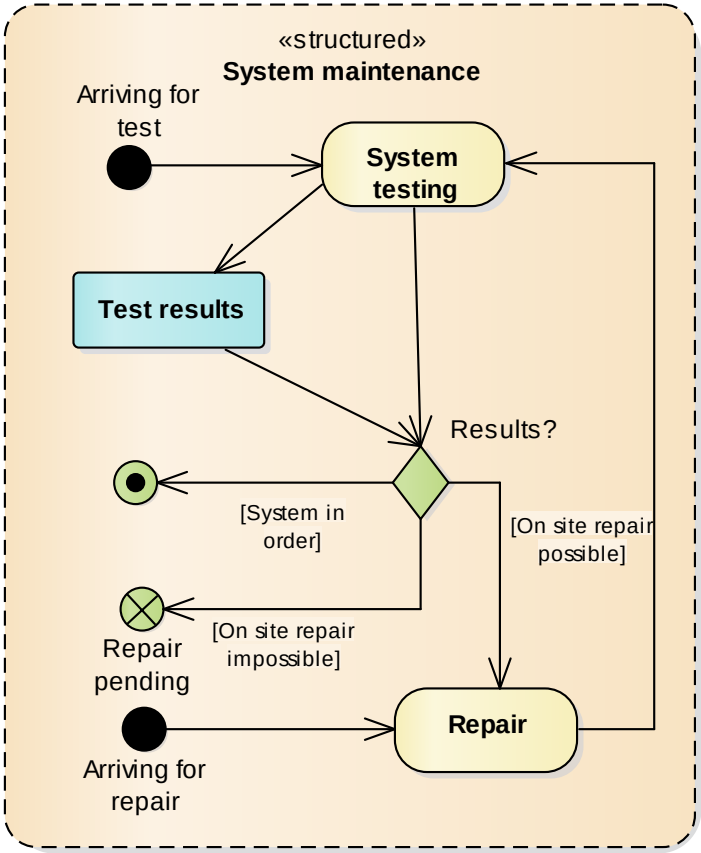
- בנוסף לזרימת הבקרה, ניתן לייצג בתרשים פעילות גם זרימת נתונים בין הפעילויות השונות
- אובייקט (Object)
  - פריט מידע בודד, המהווה קלט/פלט של פעולה
- מאגר נתונים (Data Store)
  - פעולות יכולות לאחסן בו או לשלוף ממנו פריטי מידע
- צמתי הבקרה וכללי פעולתם חלים גם לגבי זרימת מידע

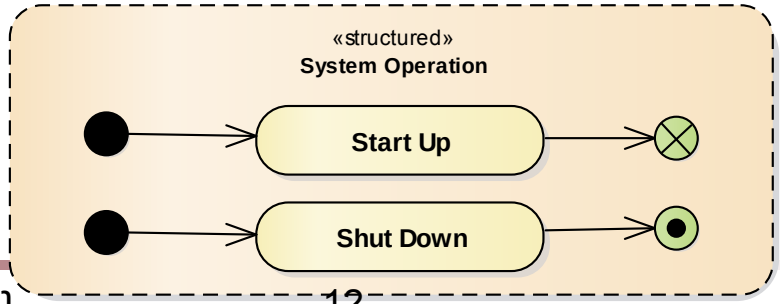
ObjectName

<<datastore>>
DataStoreName

מערכת המעליות: לוגיקת תפעול + זרימת נתונים
control flow + data flow







מטלה: הגדרת הלוגיקה העסקית

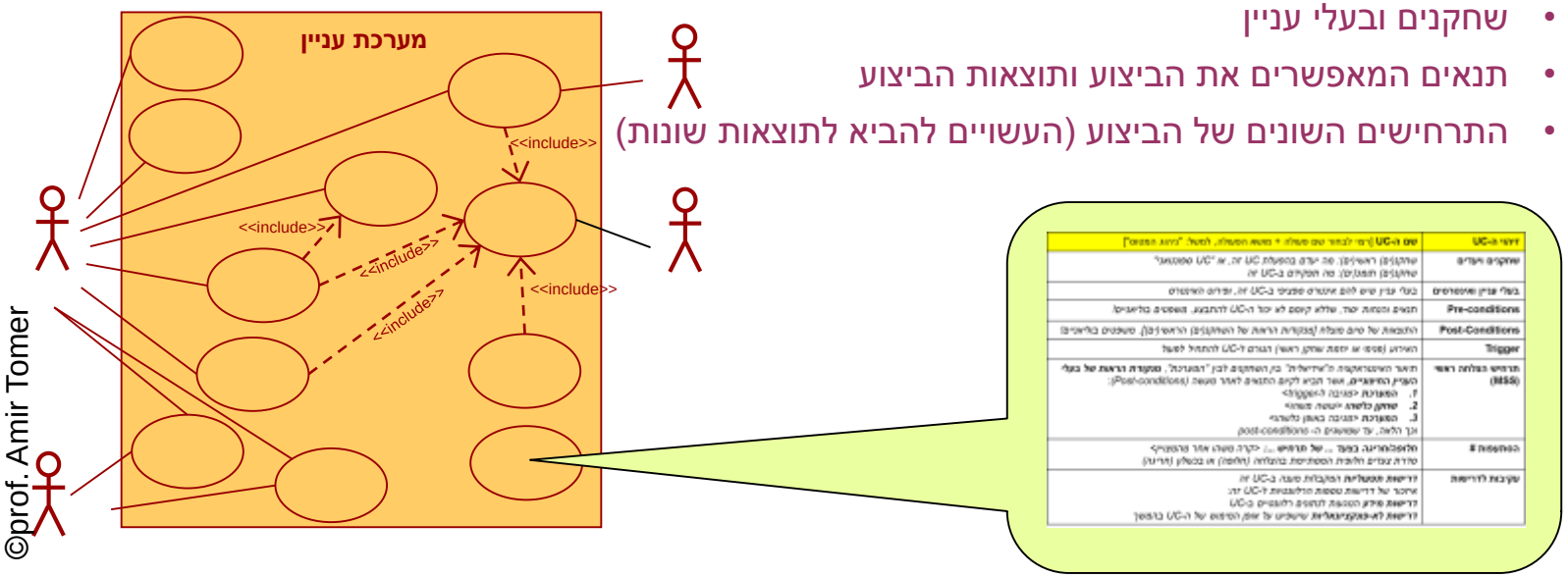
- זהו את תרחישי הלוגיקה העסקית של ePark
  - רמז: מיהם המשתמשים, מה מטרותיהם וכיצד הם יכולים להשיגן?
- ערכו תרשים פעילות (Activity Diagram) לתיאור הלוגיקה העסקית

תובנות לגבי תרשים פעילות

- התרשים מגדיר סדר פעולות אופייני / מותנה, אך לאו דווקא רצף מחייב
  - ניתן להזמין מעלית ולא לנסוע בה
- כל אחת מהפעולות המוצגות בתרשים יכולה להוות תרחיש בפני עצמה
  - תרחיש בדיקה של כלל המערכת
    - פתיחת דו"ח, בדיקת מעליות, בדיקת קומות, סיכום דו"ח
- התרשים אינו משקף את כל המידע לגבי תרחיש
  - מי המשתמשים / הגורמים המשתתפים בו?
  - האם הוא משרת אינטרסים של גורמים נוספים, חוץ מאשר המשתמשים?
  - מהו האירוע הגורם לו להתחיל לפעול?
  - תחת אלו תנאים הוא יכול להתבצע?
  - מה יקרה בסיום התרחיש, שלא היה קיים קודם?

Use Case (UC) Model: מודל לתיאור תרחישים פונקציונאליים

- Use Case Diagram
  - תרשים המתאר את המערכת בסביבתה, המייצג את
    - גבולות מערכת העניין (מלבן תוחם)
    - הסביבה החיצונית ("השחקנים")
    - שירותי המערכת (Use Cases), המבוצעים תוך אינטראקציה עם השחקנים (אליפסות)
    - הקשרים בין UCs לשחקנים ובין UCs לבין עצמם
- Use Case Specification
  - טקסט מובנה, המפרט כל UC בנפרד
    - שחקנים ובעלי עניין
    - תנאים המאפשרים את הביצוע ותוצאות הביצוע
    - התרחישים השונים של הביצוע (העשויים להביא לתוצאות שונות)



זיהוי ה-UC	שם ה-UC (יש לבחור שם משלב + פועל המתאר, למשל: "ניהול הזמנות")
שחקנים ועדים	שחקנים ראשיים: מה יעדם בפעולת UC זה, או "UC מסוגאר" שחקנים תומכים: מה הקשרים ב-UC זה
בעלי עניין ואינטרסים	בעלי עניין שיש להם אינטרס מסוגים ב-UC זה, ומרחב האינטרס
Pre-conditions	תנאים והנחות יסוד, שללא קיומם לא יכול ה-UC להתבצע, המשמשים בתיאורים
Post-Conditions	התכונות של מיפה מעלה (מצבו הראוי של האחסון) הראשי(ים), משמשים בתיאורים
Trigger	האירוע (מסוג או יחידת שתקן ראשי) הגורם ל-UC להתחיל לשמש
תרחיש בסלוחה ראשי (MSS)	תיאור האינטראקציה "אידיאלית" בין המשתמש לבין "המערכת", **מנקודת הראות של בעלי העניין החיצוניים**, אשר הביא לקיום התנאים לאחר פעולה (Post-conditions): 1. **הפעולה יוצרת** ל-Trigger 2. **שתקן בלשון יישומה מסוגה** 3. **הפעולה יוצרת** באופן מלאכתי וכן הלאה, עד שמגיעים ל- post-conditions
הרחבות #	מלאכתיריגה בפעל ... של תרחיש ... <קבוצה משלב אחר מרחבתי> מרוה עצמים ולוגיות הסמיכיות בהמלוה (חלמה) או בטעלן (מרינו)
עקבות לדרישות	**דרישות מסוגלות המקבלות מעבר ב-UC זה** איכות של דרישות מסמות הלוגבות ל-UC זה; **דרישות מיקום המבוצע תחומים רלוונטיים ב-UC** **דרישות לא-פונקציונאליות** שישפיע על אופן הביצוע של ה-UC בנוסף

בעלי עניין תפעוליים

- **בעל עניין תפעולי** (Operational Stakeholder) הינו אדם, צוות או ארגון אשר יש לו אינטרסים או נושאי עניין (concerns) הנוגעים **לפעולת המערכת**
  - השירותים אותם הוא מצפה לקבל ממנה, או שהוא מצופה לספק לה
  - ביצוע פעולות נוספות לצרכיו, גם תוך כדי מתן שירותים לבעלי עניין אחרים
  - איכות תפעול המערכת תוך מתן שירותיה
  - השפעת התפעול של המערכת עליו, תוך כדי פעולה ולאחריה
  - אופן עמידת תפעול המערכת בסטנדרטים/רגולציות אותם הוא מייצג
  - ...
- בעלי עניין תפעוליים אופייניים
  - משתמשים
  - טכנאים / מתחזקים
  - אדמיניסטרטורים
  - רגולטורים / מומחים שיש להם עניין באופן בו המערכת עובדת
    - למשל: בטיחות פונקציונאלית, ארגונומיה, אבטחת מידע
  - דרגי ניהול / פיקוד
    - עניינם במידע שעל המערכת לאסוף בזמן פעולתה
- כל תרחיש תפעולי נוגע לתת-קבוצה של בעלי העניין התפעוליים
  - האינטרסים שלהם משתנים מתרחיש לתרחיש

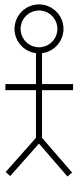
שירות המעליות – בעלי עניין תפעוליים והאינטרסים שלהם

בעל עניין	אינטרסים [תועלות, מאפייני איכות]	תפקיד
נוסע	הגעה מקומה לקומה באופן מהיר ובטוח [שימושיות, ביצועים, אמינות, זמינות]	משתמש
טכנאי	ביצוע יעיל ואמין של תחזוקה [בדיקתיות]	משתמש
מחלץ	חילוץ אמין ובטוח של נוסעים שנתקעו [זמינות]	מפעיל
תקן בטיחות	שמירה על שלום המשתמשים [בטיחות]	-

שחקנים

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

סימון UML



שם השחקן

האם פורץ הוא שחקן של מערכת אזעקה?

- פורץ מפעיל את מערכת האזעקה, ולכן ניתן לחשוב שהוא שחקן, אבל...
- יש דברים נוספים שעלולים להפעיל את מערכת האזעקה
  - חתול
  - וילון
  - ברק
  - לכלוך על גלאי
- האם כל אלה הם שחקנים?
  - ואם לא, אז מיהו השחקן של מערכת האזעקה?

- התשובה:
  - שחקן הוא ישות **המשתפת פעולה** עם המערכת **ומודעת לאינטראקציה** שלה איתה
    - השחקן והמערכת הם מרכיבים של הארגון, ולכן פועלים במשותף
  - מערכת האזעקה לא מופעלת ע"י שחקן חיצוני אלא ע"י אירוע פנימי
    - החלטה של המערכת על פי מצב הגלאי/ם

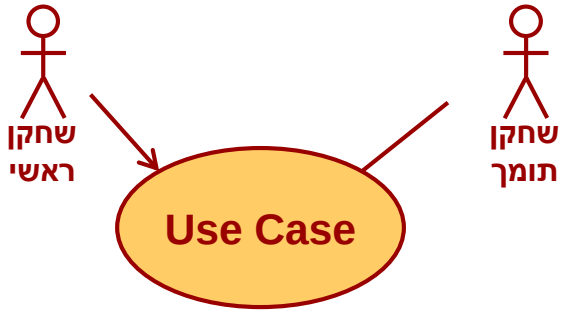
Use Cases

- **Use Case (UC)** הינו מטלה **שלמה** המבוצעת על ידי המערכת מתוך **מטרה** להפיק **תוצאות מוגדרות ובנות-מדידה** עבור **בעל-עניין** אחד או יותר
  - UC יכול להתבצע ביוזמת שחקן או ביוזמת המערכת עצמה
    - UC המתבצע ביוזמת המערכת, כתוצאה מאירוע או תנאי בתוכה, נקרא **ספונטאני** או **פנימי**
  - UC הוא תיאור מוסכם של אינטראקציה בין המערכת לשחקניה
    - האינטראקציה ו/או תוצאותיה **ניכרות** (visible) לבעלי העניין
  - UC כולל את כל האינטראקציות האפשריות שניתן לבצע לצורך השגת אותן תוצאות, **בין אם תוצאות אלה הושגו ובין אם לאו**
  - מספר מופעים של אותו UC יכולים להתבצע במקביל

סימון UML

Use Cases ושחקנים

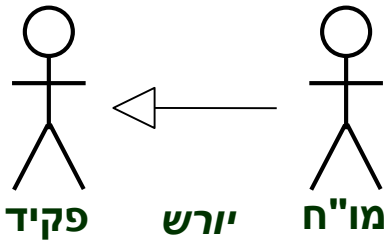
- כל UC משוייך לקבוצת שחקנים, משני סוגים:
  - **שחקן ראשי (Primary Actor)** הינו שחקן היכול **ליזום** (להפעיל) ביצוע של UC
    - הייזום נעשה דרך אחד **הממשקים המסופקים*** של המערכת
    - כאשר יש יותר משחקן ראשי אחד המשמעות היא **שכל אחד** מהם יכול ליזום את ביצוע ה-UC באופן **בלתי תלוי**
    - שחקן ראשי יוזם ביצוע של UC כדי להשיג **יעד** כלשהו
  - **שחקן תומך (Supporting Actor)** הינו שחקן (שאיננו ראשי) אשר המערכת מבצעת איתו אינטראקציה תוך כדי ביצוע של UC
    - האינטראקציה מתבצעת ביוזמת המערכת דרך אחד **הממשקים הנדרשים*** שלה
    - תפקידו של שחקן תומך הוא **לסייע** למערכת להשיג את מטרתה



* המונחים ממשק מסופק וממשק נדרש יוגדרו בהמשך הקורס

עוד על Use Cases ושחקנים

- UC ללא שחקנים ראשיים הוא UC ספונטאני (פנימי)
- 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



- פרט לשחקנים, משוייך כל UC לקבוצה של בעלי עניין רלוונטיים
  - **בעל עניין רלוונטי (של UC נתון)** הינו בעל עניין של המערכת שיש לו **אינטרס(ים) ספציפי(ים)** בביצוע של **UC זה**

מרכיבי Use Case Specification ומהלכו

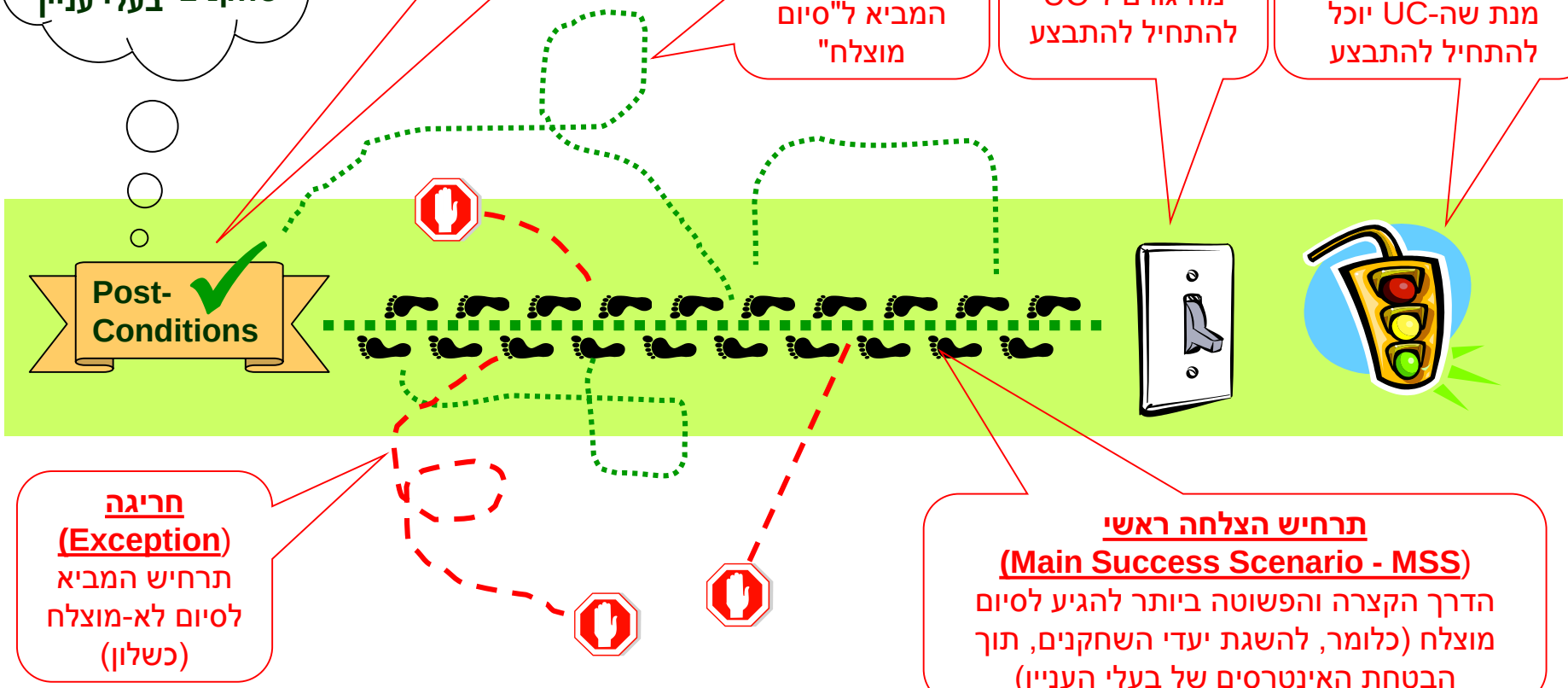
"סיום מוצלח"
הושגו יעדי השחקנים והובטחו האינטרסים של בעלי העניין

תנאים לאחר-מעשה (Post-conditions)
התוצאות של "סיום מוצלח"

חלופה (Alternative)
תרחיש שונה המביא ל"סיום מוצלח"

אירוע מפעיל (Trigger)
מה גורם ל-UC להתחיל להתבצע

תנאים מקדימים (Pre-conditions)
מה צריך להתקיים על מנת שה-UC יוכל להתחיל להתבצע



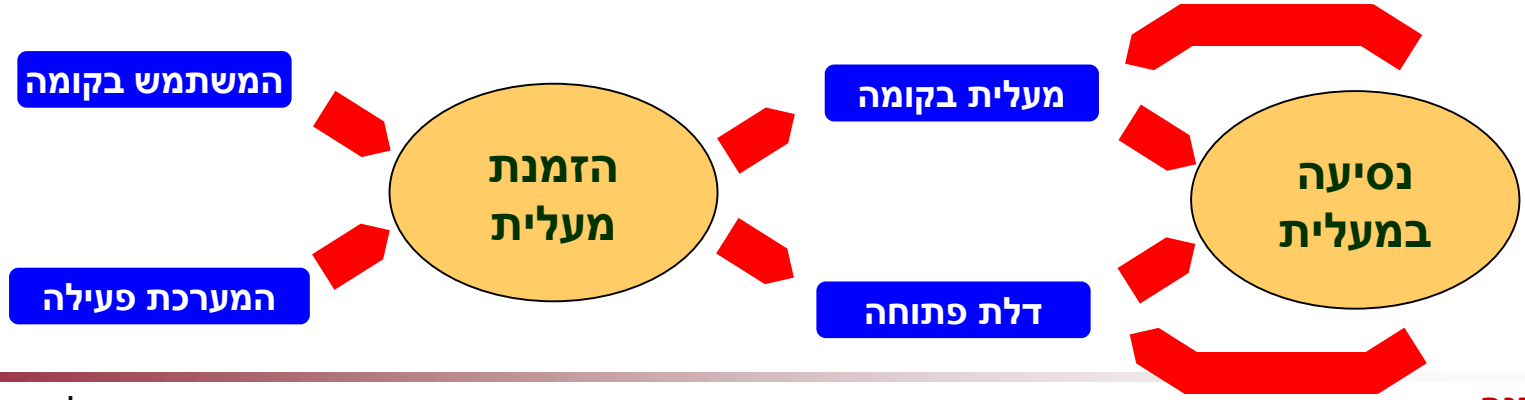
חריגה (Exception)
תרחיש המביא לסיום לא-מוצלח (כשלון)

תרחיש הצלחה ראשי (Main Success Scenario - MSS)
הדרך הקצרה והפשוטה ביותר להגיע לסיום מוצלח (כלומר, להשגת יעדי השחקנים, תוך הבטחת האינטרסים של בעלי העניין)

קשרים מותנים בין Use Cases

- Post-conditions של UC אחד יכולים להוות Pre-conditions של UC(s) אחר(ים)
  - לדוגמה, כספומט

- זרימת הפעילות איננה בהכרח מוגדרת מראש (ההפעלה היא נסיבתית – כאשר נוצרו התנאים ניתן לפעול)
- ה- Pre-conditions יכולים להתקיים כתוצאה מהפעלת UC אחר
  - לדוגמה, מעלית



תבנית לכתיבת מפרט טקסטואלי של Use Case

זיהוי ה-UC	**שם ה-UC** [רצוי לבחור שם פעולה + מושא הפעולה, למשל: "ניהוג המטוס"]
שחקנים ויעדים	*שחקנ(ים) ראשי(ים): מה יעדם בהפעלת UC זה, או "UC ספונטאני"* *שחקנ(ים) תומכ(ים): מה תפקידם ב-UC זה*
בעלי עניין ואינטרסים	*בעלי עניין שיש להם אינטרס ספציפי ב-UC זה, ופירוט האינטרס*
Pre-conditions	*תנאים והנחות יסוד, שללא קיומם לא יכול ה-UC להתבצע. משפטים בוליאניים!*
Post-Conditions	*התוצאות של סיום מוצלח [מנקודות הראות של השחקנ(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 שמושגים ה- post-conditions*
הסתעפות #	***חלופה/חריגה בצעד ... של תרחיש ...:** <קרה משהו אחר מהמצויין>* *סדרת צעדים חלופית המסתיימת בהצלחה (חלופה) או בכשלון (חריגה)*
עקיבות לדרישות	***דרישות תפעוליות** המקבלות מענה ב-UC זה* *איזכור של דרישות נוספות הרלוונטיות ל-UC זה:* ***דרישות מידע** הנוגעות לנתונים רלוונטיים ב-UC* ***דרישות לא-פונקציונאליות** שישפיעו על אופן המימוש של ה-UC בהמשך*

תרחישים יכולים להכיל לולאות
ומשפטים מותנים (אם... אז...)
אבל **לא** הסתעפויות (אם... אז... אחרת...)

הסתעפויות יכולות
להיכתב, בשלב ראשון
באופן כללי, וניתן לפרטן
ולהרחיבן בהמשך

כתיבת Use Cases ברמת המערכת

- ניתן לכתוב מפרטי Use Cases בכל אחת מרמות הפירוק של המערכת (ארגון, מערכת, פריט, ...)
  - אנו נשתמש בטכניקה של כתיבת מפרטי Use Case מובנים עבור תרחישי הפעולה ברמת המערכת
  - מאחר ואנו מתמקדים בתוכנה, ניתן יהיה להשתמש בתרחישים גם כמפרט דרישות תוכנה (SRS = Software Requirements Specification)
- ניעזר במקורות הבאים:

1. תרשים הלוגיקה העסקית
   - ה-UC ברמת המערכת אלה הפעילויות המרכיבות את התרחישים העסקיים
2. האפיון התפעולי ("סיפור הלקוח")
   - אמור להכיל תיאורים לשימוש במערכת בתרחישים שונים

נוסע הנמצא בקומה כלשהי ורוצה להזמין מעלית לוחץ על הכפתור המתאים לכיוון הנסיעה המבוקש. אם לא היה דלוק קודם לכן, הכפתור נדלק. בעקבות הלחיצה, מעלית כלשהי הנמצאת בכיוון הנסיעה המבוקש תגיע לקומה, זמן דקה לכל היותר. עם הגעתה נפתחת הדלת והכפתור כבה.

נוסע הנמצא בתוך המעלית ורוצה להגיע לקומה כלשהי לוחץ על הכפתור המתאים. אם לא היה דלוק קודם הכפתור נדלק בעקבות הלחיצה. הדלת נסגרת, לאחר השהיה, והמעלית ממשיכה בנסיעה באותו כיוון, כאשר היא עוצרת בכל קומה שכפתורה דולק. כאשר המעלית נעצרת בקומה הדלת נפתחת והכפתור המתאים לקומה כבה.

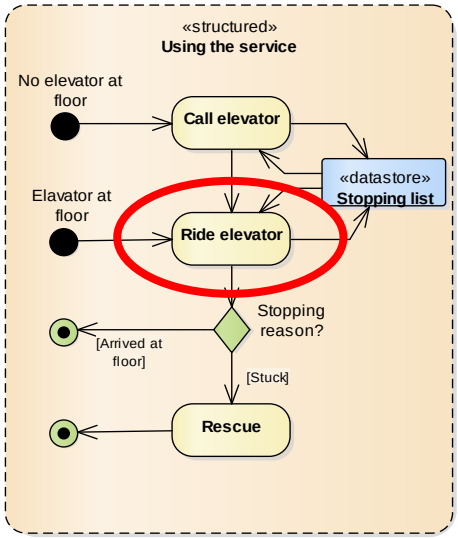
נוסע במעלית יכול לבצע את המעלית בזמן נסיעה באמצעות לחיצה על כפתור עבודת החירום. במקרה זה המעלית עוצרת מיד וכל בקשות העצירה שלה מתבטלות. לאחר מכן ניתן להחזיר את המעלית לפעולה על ידי לחיצה על כפתור עבור קומה כלשהי.

במקרה שהמעלית נתקעה במהלך נסיעה מזעיק הנוסע חילוץ באמצעות כפתור החילוץ. המחלץ (איש האחזקה של הבניין) מגיע לפאנל החילוץ שבחדר המכונות ומפעיל פקודות להורדת המעלית לקומת הקרקע ולפתיחת הדלת.

טכנאי, המגיע אחת ל-6 חודשים, יכול לבצע בדיקה מקיפה של כל המערכת ולתקן תקלות באמצעות פיקוד הטכנאי שבחדר המכונות.

מערכת המעליות תעמוד בכל תקני הבטיחות הישימים.

המערכת תונגש לבעלי מוגבלויות שונות.



SUC-1 הזמנת מעלית

SUC-1	הזמנת מעלית
שחקנים ויעדים	נוסע: לקבל מעלית זמינה לנסיעה
ב"ע ואינטרסים	אין
Pre-conditions	• הנוסע נמצא בקומה כלשהי בה נמצאת דלת של מעלית • המערכת פעילה [Post-cond. של UC "איתחול מערכת"]
Post-Conditions	• מעלית פתוחה נמצאת בקומה בה נמצא הנוסע (יעד)
Trigger	• **הנוסע** לוחץ על כפתור עליה / ירידה בקומה
MSS	1. **המערכת** קולטת את הלחיצה 2. **הכפתור** נדלק 3. **המערכת** מאתרת מעלית הנוסעת בכיוון המבוקש 4. **המערכת** מקצה את העצירה למעלית 5. **המעלית** מגיעה לקומה 6. **דלת** המעלית נפתחת 7. **כפתור** הקומה כבה
הסתעפות א'	חלופה בצעד 2 של MSS: הכפתור כבר דלוק (כבר הוקצתה מעלית) 2א1. מעבר לצעד 5
עקיבות לדרישות	...

SUC-2 נסיעה במעלית (1)

SUC-2	נסיעה במעלית
שחקנים ויעדים	נוסע: להגיע במעלית לקומה מבוקשת
ב"ע ואינטרסים	תקן בטיחות: נוסעים לא יישארו תקועים במעלית
Pre-conditions	• הנוסע נמצא בתוך המעלית • המעלית פעילה [Post-cond. של UC "איתחול מערכת"]
Post-Conditions	• המעלית הגיעה לקומה המבוקשת (יעד) • הנוסעים יכולים לצאת מהמעלית (אינטרס)
Trigger	• **הנוסע** לוחץ על כפתור קומה מבוקשת
MSS	1. **המעלית** קולטת את הלחיצה ורושמת לעצמה בקשה לעצירה בקומה 2. **הכפתור** שנלחץ נדלק 3. **דלת** המעלית נסגרת (במידה והיתה פתוחה) 4. **המעלית** ממשיכה בנסיעה לקומה הבאה בה היא נדרשת לעצור 5. **המעלית** נעצרת בקומה 6. **הדלת** נפתחת 7. **כפתור** הקומה כבה 8. חזרה לצעד 3

צעדים אלה מתבצעים למעשה במקביל לשאר התרחיש. בהמשך נראה כיצד ממשים זאת.

המשך בשקף הבא...

SUC-2 נסיעה במעלית (2)

... המשך מהשקף הקודם

הסתעפות א'	**חריגה** בצעד 4 של MSS: הנוסע לחץ על כפתור עצירת חירום 4א1. **המעלית** נעצרת מיד 4א2. **המעלית** מבטלת את כל בקשות העצירה הקיימות 4א3. התרחיש מסתיים
הסתעפות ב'	**חריגה** בצעד 4 של MSS: המעלית נתקעה 4ב2. התרחיש מסתיים
עקיבות לדרישות	...

SUC-3 חילוץ נוסע

SUC-3	חילוץ נוסע
שחקנים ויעדים	נוסע: להיחלץ ממעלית שנתקעה מחלץ: שחקן תומך
ב"ע ואינטרסים	תקן בטיחות: נוסע לא יישאר תקוע במעלית
Pre-conditions	• המעלית תקועה (נתקעה במהלך נסיעה [extended UC-2])
Post-Conditions	• הנוסע יכול לצאת מהמעלית (יעד + אינטרס)
Trigger	• **הנוסע** מזעיק חילוץ
MSS	1. **המערכת** מזעיקה מחלץ 2. **מחלץ** מגיע לחדר המכונות ומפעיל את המעלית ב-mode חילוץ 3. **המעלית** מגיעה לקומת הקרקע 4. **הדלת** נפתחת
הסתעפויות	אין
עקיבות לדרישות	...

SUC-4 בדיקת מערכת (1)

UC-4	בדיקת מערכת
שחקנים ויעדים	טכנאי: להשלים בדיקה מקיפה ולוודא שהמערכת תקינה
ב"ע ואינטרסים	נוסעים + תקן בטיחות: המערכת תקינה ובטוחה לשימוש (אמינות, זמינות, בטיחות)
Pre-conditions	• המערכת פעילה (באמצעות UC "איתחול מערכת") • המערכת לא נמצאת בשימוש
Post-Conditions	• כל פונקציות המערכת תקינות • נרשם דו"ח תקינות מלא
Trigger	• **הטכנאי** מפעיל את תוכנת הבדיקה
MSS	1. **הטכנאי** פותח דו"ח תקינות חדש 2. **הטכנאי** מזמין מעלית לעליה בקומה שטרם נבדקה והמעלית מגיעה [UC-1] 3. **הטכנאי** מזמין מעלית לירידה באותה קומה והמעלית מגיעה [UC-1] 4. **הטכנאי** מסמן בדו"ח שהקומה תקינה 5. **הטכנאי** חוזר על צעדים 2-5 עבור כל הקומות 6. **הטכנאי** נכנס למעלית שטרם נבדקה 7. **הטכנאי** נוסע במעלית לכל הקומות [UC-2] 8. **הטכנאי** מסמן בדו"ח שהמעלית תקינה 9. **הטכנאי** חוזר על צעדים 7-9 עבור כל המעליות

המשך בשקף הבא...

SUC-4 בדיקת מערכת (2)

... המשך מהשקף הקודם

הסתעפות א'	**חלופה** מצעד S (S = 2, 3 או 7) של MSS: תקלה בקומה/מעלית כלשהי S1א. **הטכנאי** מתקן את התקלה [UC-5] S2א. **הטכנאי** חוזר על הבדיקה שנכשלה ומוודא שהתקלה תוקנה S3א. התרחיש נמשך
הסתעפות ב'	**חריגה** בצעד S1א של הסתעפות א': הטכנאי לא הצליח לתקן את התקלה S1ב1א. **הטכנאי** מסמן את התקלה בדו"ח S1ב2א. עבור לצעד 5 או 9, בהתאמה
עקיבות לדרישות	...

SUC-6 איתחול מערכת

SUC-6	איתחול מערכת
שחקנים ויעדים	איש האחזקה: הכנסת מערכת המעליות לפעולה
ב"ע ואינטרסים	נוסעים: המערכת זמינה לשימוש
Pre-conditions	• המערכת איננה פועלת
Post-Conditions	• המערכת פעילה ומאפשרת שימוש
Trigger	• **איש האחזקה** מדליק את המערכת
MSS	1. **המערכת** מאתחלת את כל רכיביה (מעליות, פאנלים וכו') 2. **המערכת** מוודאת שכל מרכיביה תקינים 3. **המערכת** מציגה חיווי תקינות
הסתעפות א'	**חריגה** בצעד 1 של MSS: לא כל מרכיבי המערכת תקינים 1א1. **המערכת** מציגה חיווי אי-תקינות התרחיש מסתיים
עקיבות לדרישות	...

עקיבות הדרישות התפעוליות למודל UC

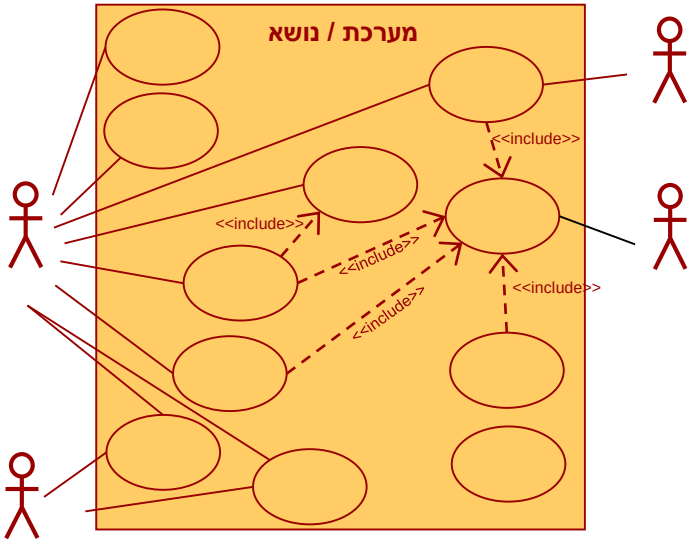
SUC	סוג	נוסח	זיהוי
		. . .	
SUC-1	OR	נוסע הנמצא בקומה כלשהי ... לוחץ על הכפתור המתאים לכיוון הנסיעה המבוקש	
SUC-1	OR	כפתור קומה נדלק בעקבות לחיצה, אם לא היה דלוק קודם לכן	
SUC-1	OR	בעקבות לחיצה על כפתור בקומה, מעלית כלשהי הנמצאת בכיוון הנסיעה המבוקש תגיע לקומה	
SUC-1	OR	עם הגעת מעלית לקומה נפתחת הדלת וכפתור הקומה כבה	
		. . .	
SUC-2	OR	נוסע הנמצא בתוך המעלית ורוצה להגיע לקומה כלשהי לוחץ על הכפתור המתאים	
SUC-2	OR	כפתור מעלית נדלק בעקבות לחיצה, אם לא היה דלוק קודם	
SUC-2	OR	כאשר המעלית נעצרת בקומה הדלת נפתחת והכפתור המתאים לקומה כבה	
SUC-3	OR	במקרה שהמעלית נתקעה במהלך נסיעה מזעיק הנוסע חילוץ	
SUC-4	OR	... המערכת נבדקת... בידי טכנאי מוסמך	
SUC-5	OR	אם הטכנאי מגלה תקלה הוא מנסה לתקן אותה...	

מטלה: כתיבת מפרט UC

- כתבו מפרטי Use Case טקסטואליים עבור ה-UC הבאים:
  - הרשמה וכניסה לפארק (Check In)
  - כניסה למתקן (Enter Device)
  - ניטור מתקנים (Device Monitoring)
- השתמשו בתבנית לכתיבת מפרטי Use Case

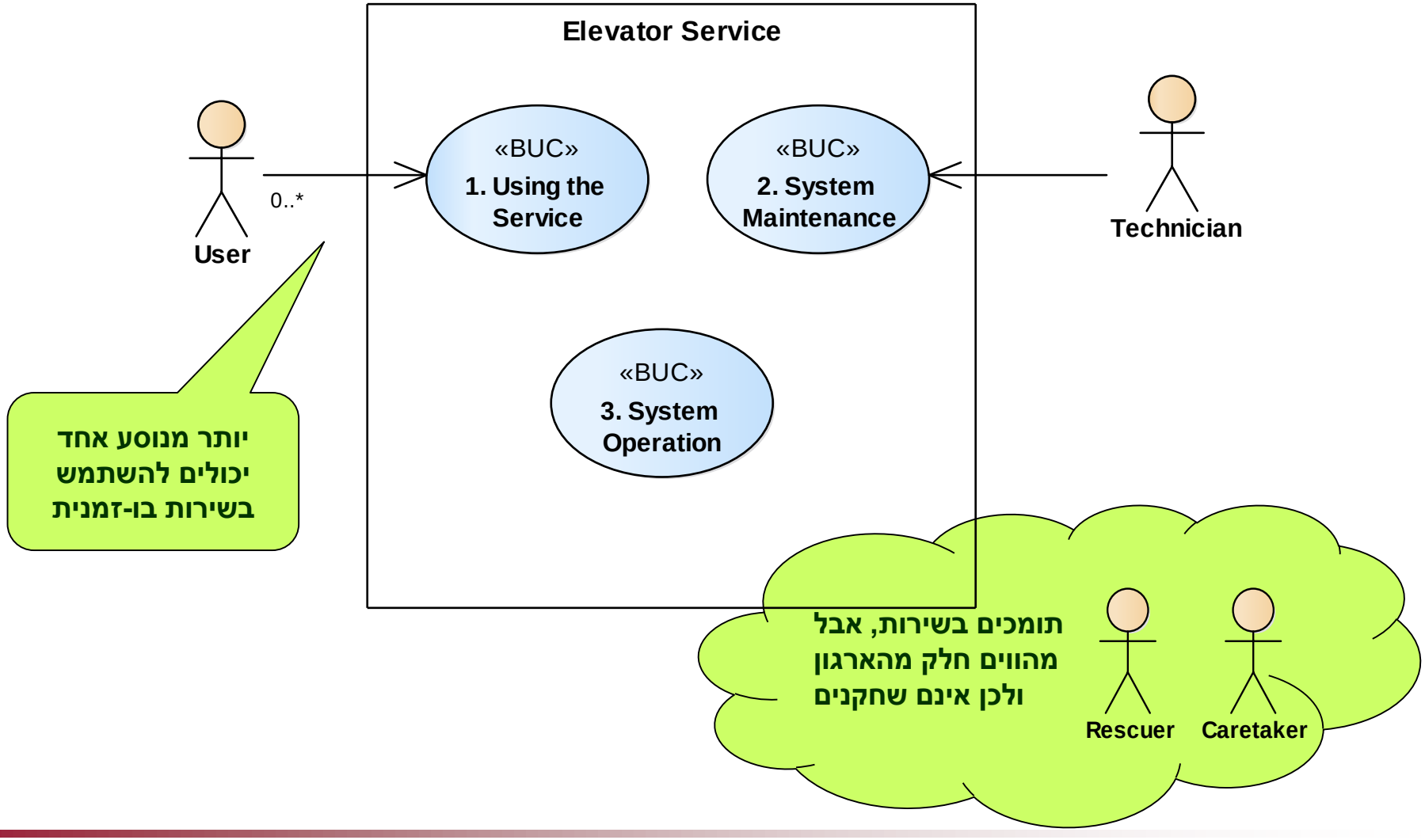
תרשים Use Cases (Use Case Diagram)

- תרשים Use Case (UCD) הינו ייצוג גראפי של קבוצת Use Cases וקשריהם עם קבוצת שחקנים
  - למרות הקשר ההדוק בין Use Cases ובין בעלי עניין שאינם שחקנים, אין ב-UML ייצוג גראפי שלהם
- התרשים כולל את המרכיבים הבאים:
  - **מלבן תוחם**, המייצג את גבולות מערכת-העניין / הנושא (System Boundary / Subject)
  - **אליפסות** בתוך הגבולות, המייצגות Use Cases
  - **דמויות** מחוץ לגבולות, המייצגות שחקנים
    - כל שחקן מקושר ל-UC אחד או יותר
  - **חיצים** מקווקוים, המייצגים יחסים בין UCs
    - יוסברו בהמשך

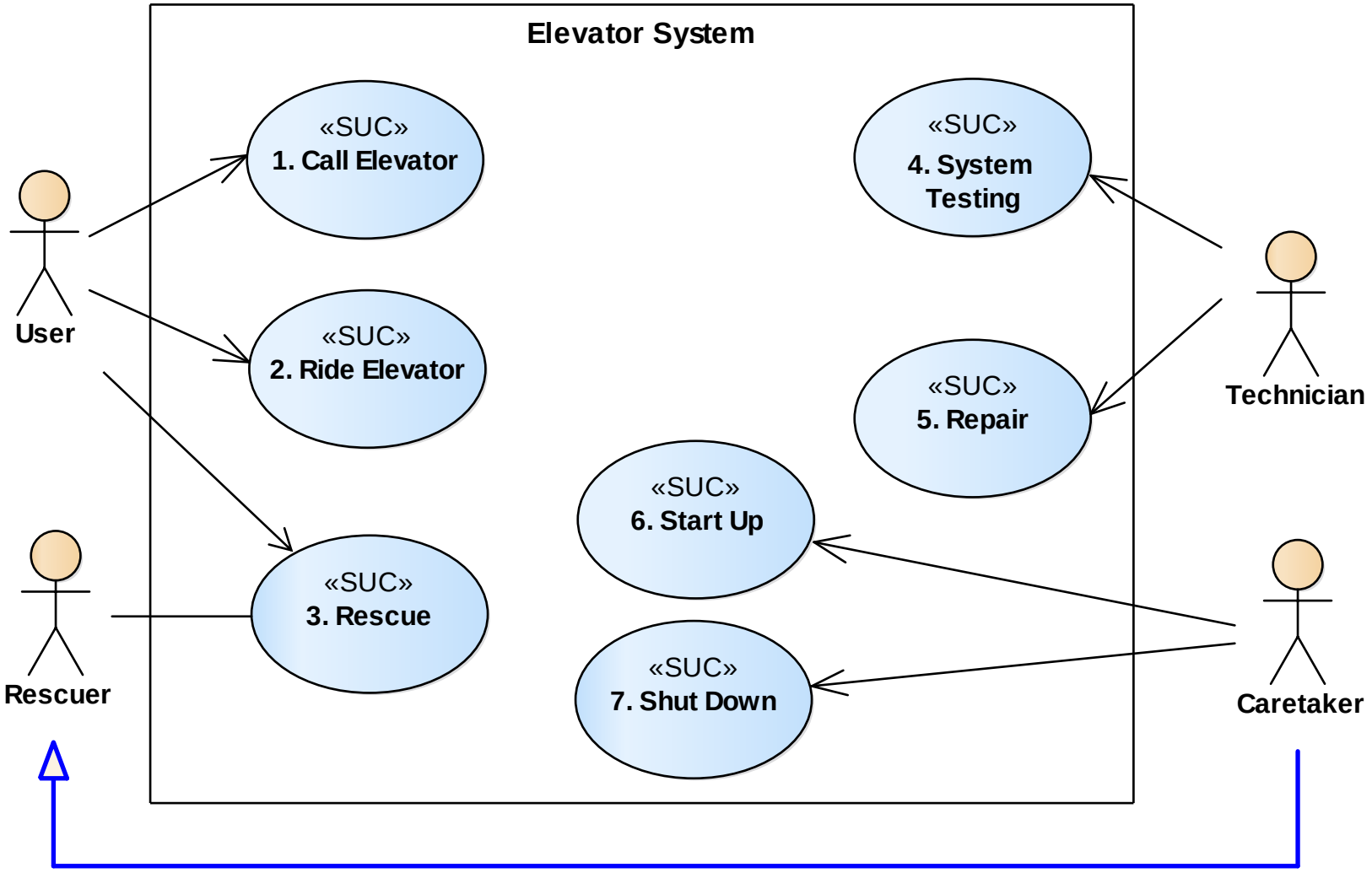


- **למרות שתרשים ה-UC נועד לייצג את תפעול המערכת, הוא עצמו איננו מודל דינאמי!**

Use Case Diagram ברמת הארגון

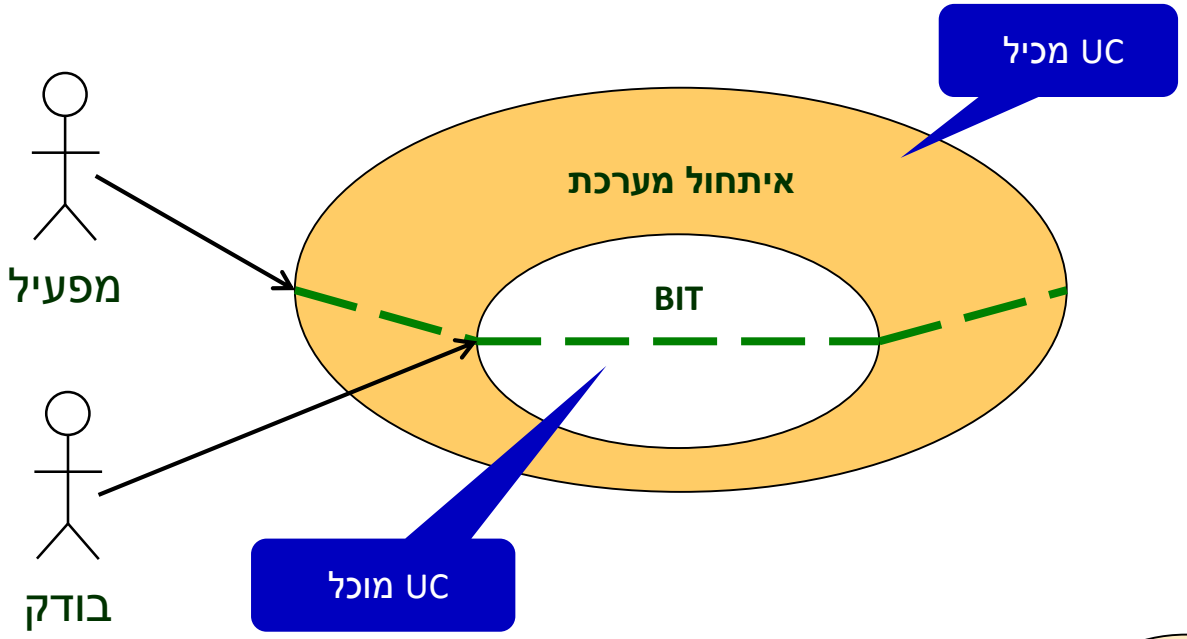


UCD (בסיסי) ברמת המערכת

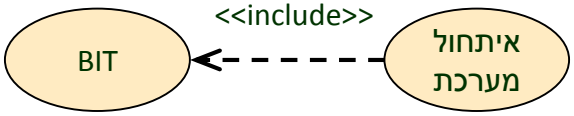


תלות "הכלה" (<<include>>) בין Use Cases

- Use Case A **מכיל את** Use Case B אם B הוא חלק אינטגראלי מ-A
  - B עדיין יכול להיות מופעל באופן עצמאי, ע"י שחקנים אחרים
    - לדוגמה BIT (Built in Test)



סימון UML

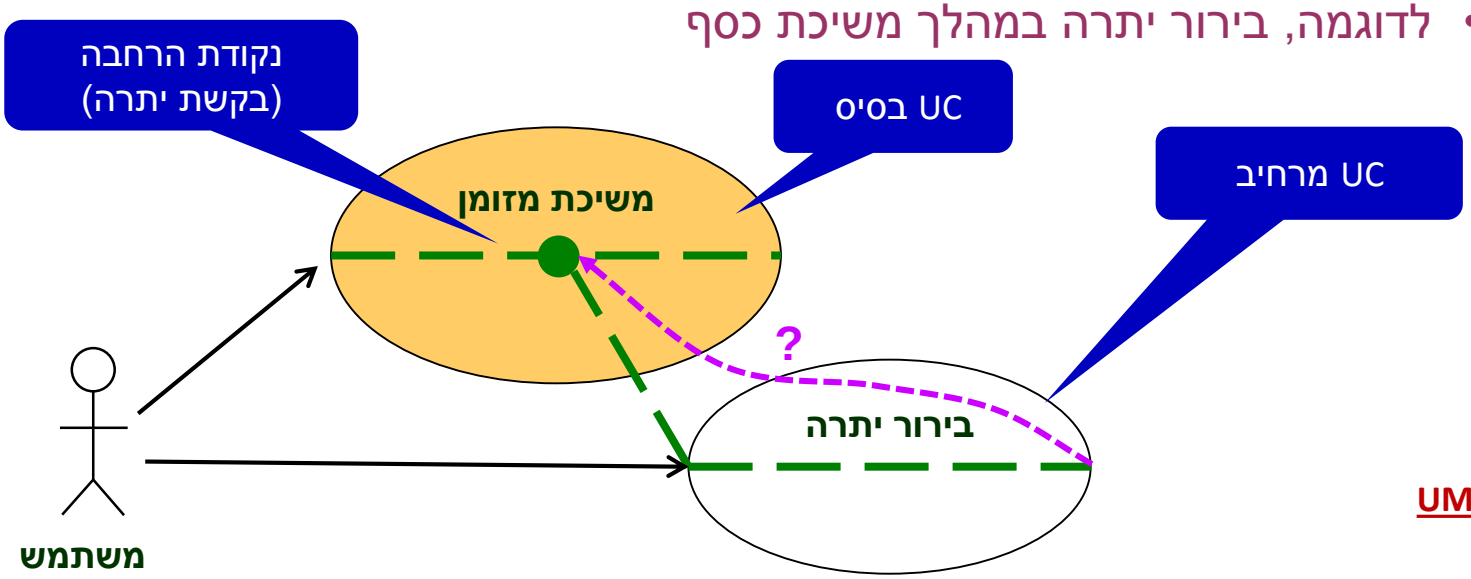


מתי להשתמש ב- <<include>>?

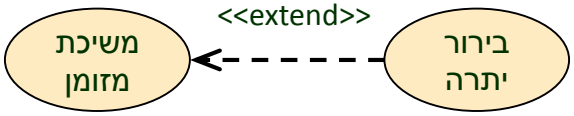
- כדאי להפריד חלק מ-UC ולהגדירו כ-UC עצמאי באחד או יותר מהמקרים הבאים:
  1. ה-UC המוכל הוא גדול (כולל מספר רב של פעולות)
  2. ההתנהגות ה"מוכלת" משותפת ליותר מ-UC אחד (ואז היא תהיה מוכלת בכולם)
  3. ניתן להפעיל את ה-UC המוכל בנפרד מ-UC הבסיס

תלות "הרחבה" (<<extend>>) בין Use Cases

- Use Case B **מרחיב את** Use Case A כאשר B ניתן להפעלה **אופציונאלית** במהלך ביצוע A
  - B מופעל ב"נקודת הרחבה" (extension point) מוגדרת בתוך A
  - B עדיין יכול להיות מופעל כ-UC עצמאי ע"י שחקנים אחרים
    - לדוגמה, בירור יתרה במהלך משיכת כסף



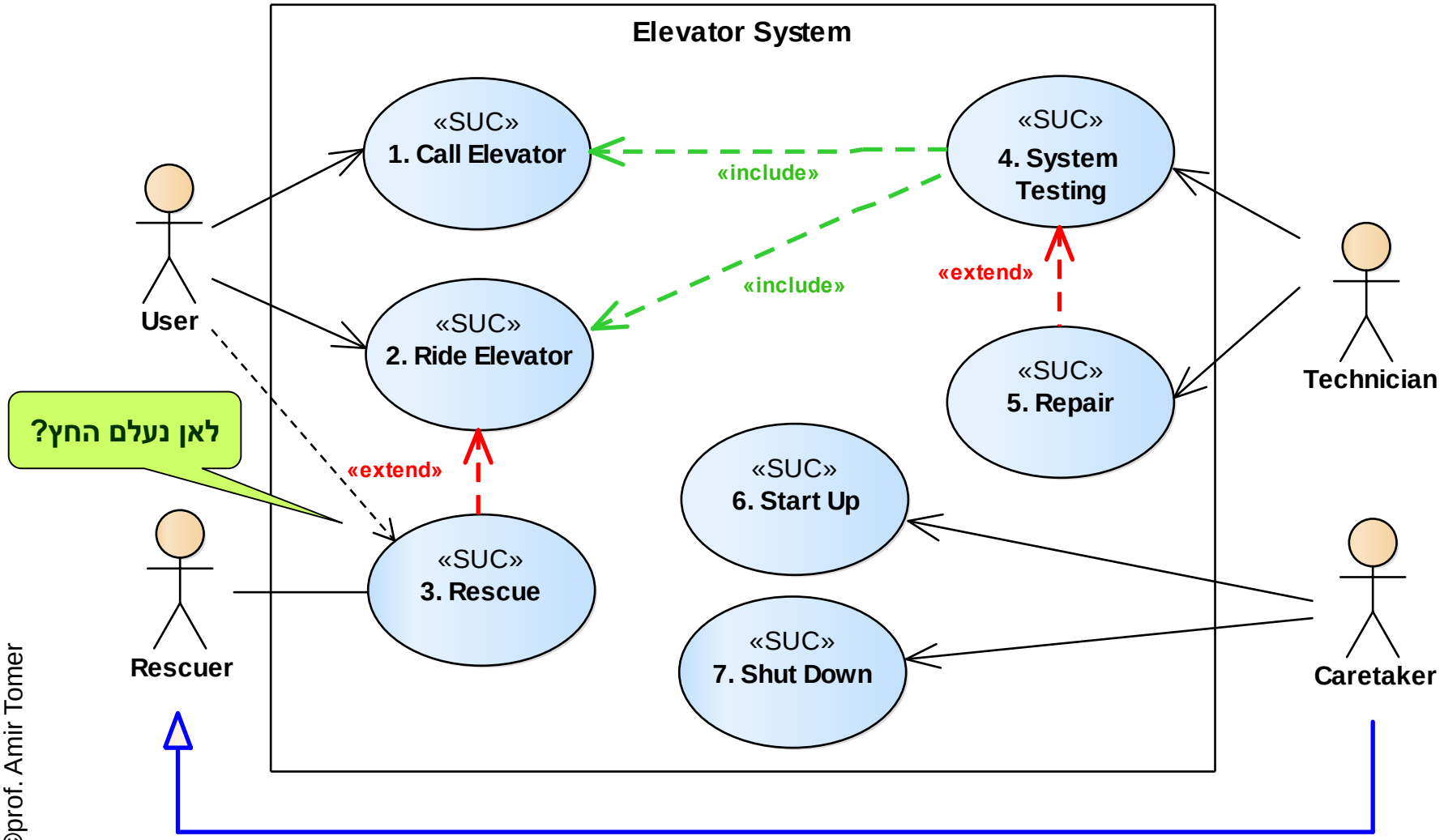
סימון UML



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

UCD למערכת המעליות עם תלויות בין UCs



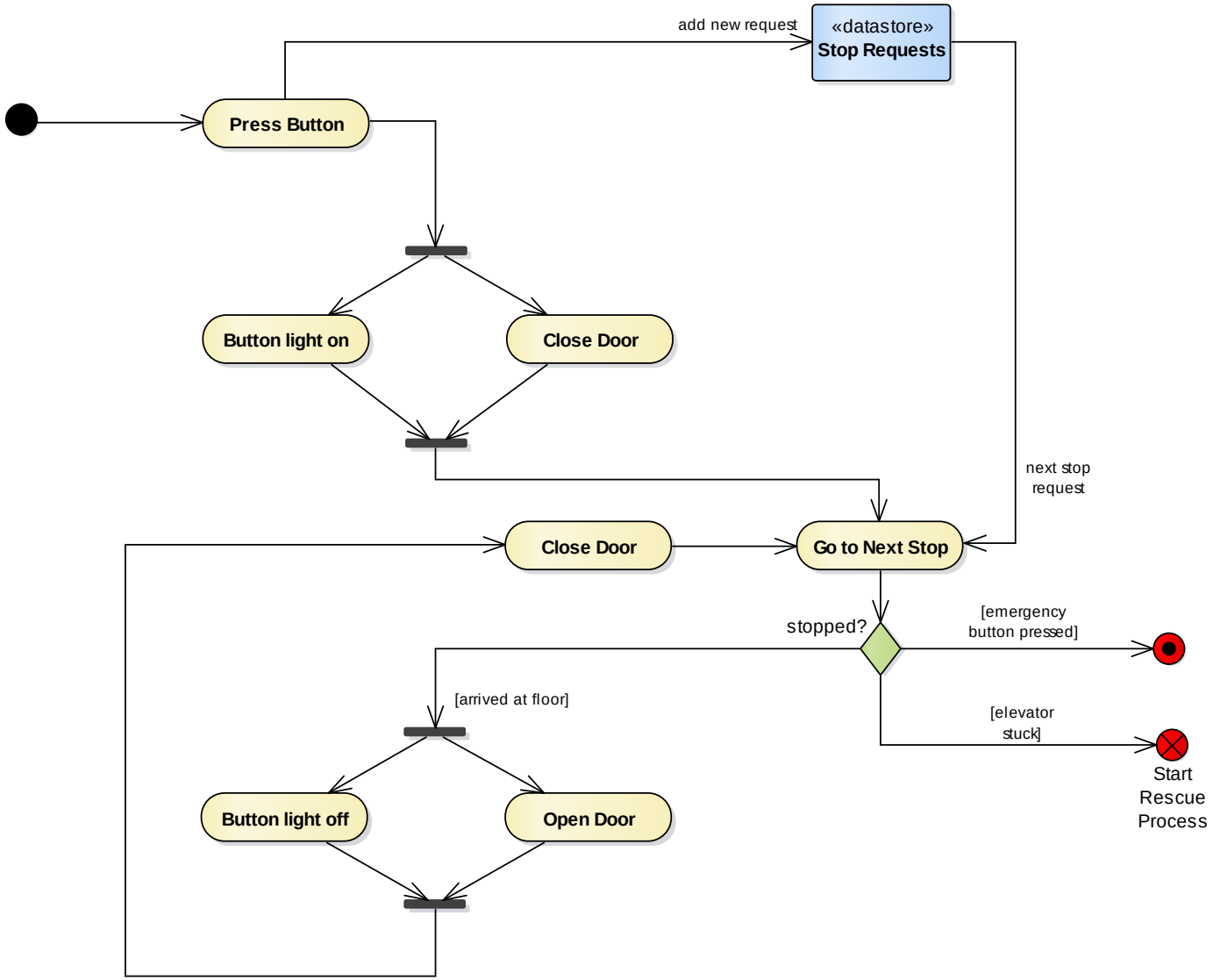
מטלה: בניית תרשים UC

- בנו Use Case Diagram למערכת ePark
  - בשלב ראשון התבססו על הפעילויות שבתרשים הלוגיקה העסקית
  - בשלב שני, נסו לראות האם יש הצדקה להוציא חלקים מ-UC מסויימים ל-UC נפרדים עם תלויות extend/include.

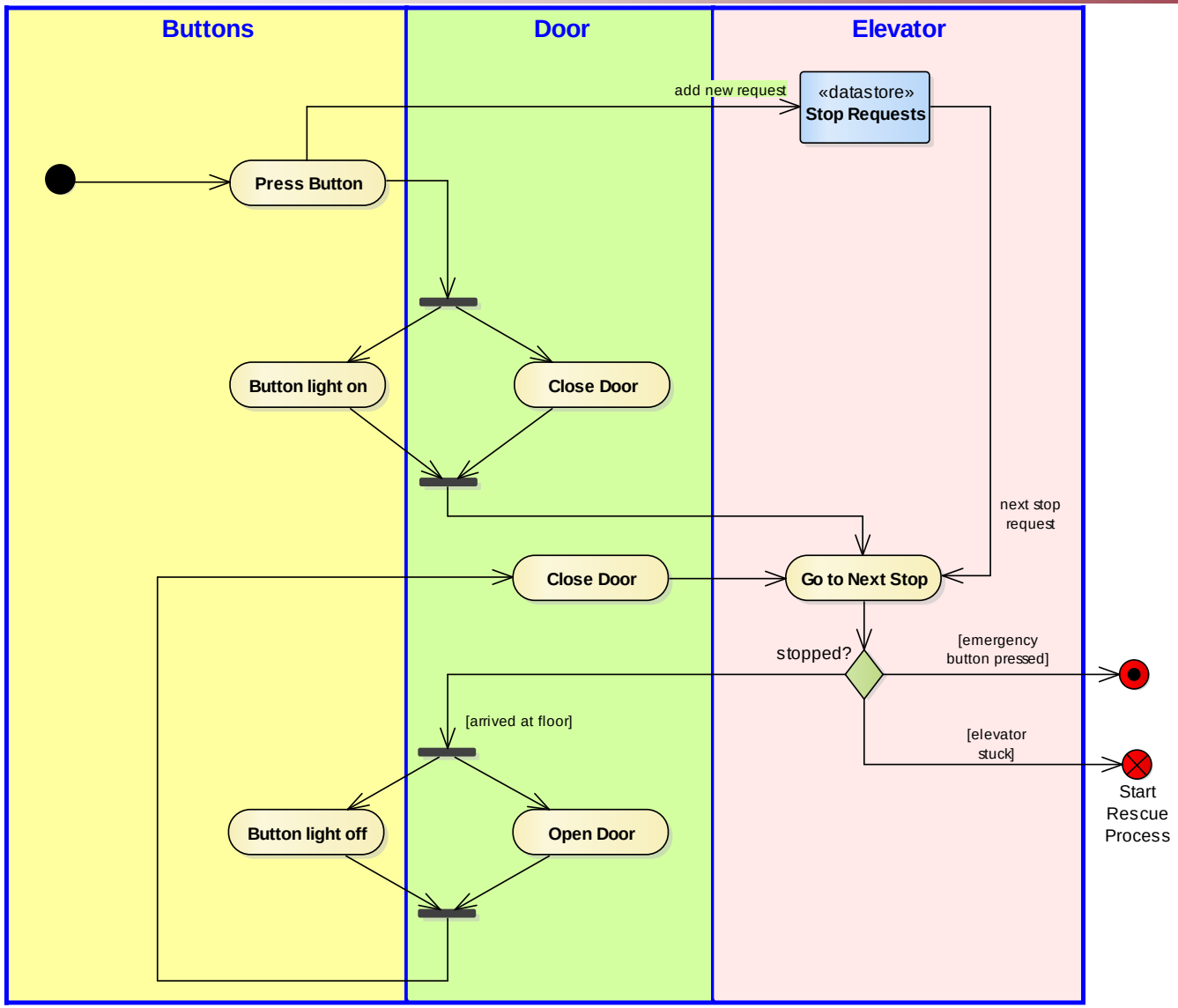
פירוט תרחישי UC באמצעות Activity Diagram

- ניתן להשתמש בתרשים פעילות (Activity Diagram) לתיאור התרחיש(ים) של use case באופן גרפי
  - נקודת המבט משתנה:
    - מפרט טקסטואלי – מנקודת ראות המשתתפים (אינטראקציה)
    - Activity Diagram – מנקודת ראות המערכת (process flow)
  - ניתן לכלול בתרשים אחד הן את תרחיש ההצלחה הראשי והן את ההסתעפויות
  - ניתן לבטא בתרשים את "חלוקת העבודה" בין המשתתפים באמצעות סימון "מסלולי שחיה" (swim lanes)
  - <u>לתשומת לב</u>: התרשים מחליף אך ורק את האירוע המפעיל + תיאור התרחישים
    - את כל שאר חלקי ה-UC (pre-cond, post-cond וכו') יש לפרט בטקסט!

SUC-2 נסיעה במעלית – Activity Diagram



הקצאת פעילות באמצעות "מסלולי שחייה" (Swim Lanes)



מטלה: תיאור תרחישים בתרשים פעילות (Activity Diagram)

- ערכו תרשימי פעילות עבור ה-UC הבאים:
  - כניסה למתקן
    - כולל זרימת בקרה ונתונים
  - מעקב אחרי ילד
    - ללא זרימת נתונים, אך עם מסלולי השחייה הבאים:
      - Central DB
      - Guardian Work-Station
      - Bracelet