

מטלה שניה: ניתוח ותכן המערכת ובניית אבטיפוס

כללי

בהתבסס על הניתוח הראשוני ותרחישי השימוש שהכנתם בשלב הראשון, עליכם להשלים את תהליך ניתוח הדרישות וליצור את תכן המערכת (Design). בהתאם לפעילות זו יש להכין מודל UML המייצג את מבנה מערכת התוכנה כפי שתיבנה בהמשך ואת אופן פעולתה המלא בהתאם לדרישות.

- **הערה** להבהרה והדגשה: תפקיד מודל התכן (Design) הוא **לתאר** את המתווה למימוש מערכת התוכנה ("software system") שאתם מפתחים: איך היא בנויה ואיך היא פועלת. לא ניתן **לתאר** את המערכת באמצעות ה"מערכת" עצמה. לכן הישות בשם: "system" אינה רלבנטית בכל צורה ואופן באף אחת מהדיאגרמות הכלולות במודל.

בנוסף, עליכם לבנות אב-טיפוס יישומי (עובד) של ארכיטקטורת המערכת הבסיסית.

- ❖ יש להסתמך על הגרסה הסופית של תיאור המערכת והגדרת הפרויקט המפורסמת במסמך "Semester Project" ב-Moodle.

הגדרות המטלה

1. המודל (שיבנה באמצעות גרסה 17.2 של Visual Paradigm) יכיל את הדיאגרמות (סה"כ 60%):

- **Activity** (18%): תיאור ביצוע התהליכים הבאים באמצעות מערכת התוכנה, כולל הקצאת פעילויות לאובייקטים בטכניקת Swim-Lanes (המלצה: דיאגרמה נפרדת לכל תהליך):
 - חנית רכב (כולל הזמנה מסירה וקבלה)
 - יצירת דו"ח זמני חניה
 - **Class** (30%): תיאור **מבנה מערכת התוכנה** + שימוש בתבנית ECB (ECB Pattern),
 - יש לפרט מתודות ספציפיות הנכללות במחלקות השונות (אין צורך לתאר את "הפעולות הבסיסיות").
 - יש לכלול את מחלקות ה-OCSF והקשר/הממשק אליהן, וכן ייצוג לחבילות הממוחזרות (JDBC, JavaFX) והקשרים אליהם.
 - יש לכלול תיאור של לפחות Design Pattern אחד הנכלל במערכת.
 - יש לציין **בהערה** במודל היכן בדיוק במודל ה-pattern מתואר.
 - **Sequence** (12%): תיאור מהלך הרצפים הבאים במערכת (המלצה: דיאגרמה לכל רצף):
 - חנית רכב (כולל מסירה וקבלה)
 - יצירת דו"ח זמני חניה
 - יצירת וצפיית דו"חות סטטיסטיות קשורות להשוואה בין סטטוס מנויים בתאריכים שונים
 - **Package** (5%): יש לכלול תיאור ארכיטקטורת התוכנה באמצעות Package Diagram.
 - במידת הצורך – השלמה עצמית של לימוד הנושא בעזרת Google:
 - לחפש: Package diagram guidelines, ולהשתמש בהגדרות והדוגמאות ב-Lucidchart או ב-Visual Paradigm.
- 🚩 **במטלה 3** תוערך איכות מודל זה מול מימוש ה-Packages בקוד והתיעוד שלהם.

כל המלל (שמות, תיעוד, וכו') במודלים יהיה באנגלית

2. (25%) **אב-טיפוס** פועל של המערכת (בשני קבצי Jar להרצה), שיממש את היכולות הבאות:

1. שרת (server) בסיסי שמתחבר ל-database שלדי שכולל את טבלה אחת הקשורה למערכת.
2. לקוח (client) רץ (במחשב נפרד מהשרת) מול השרת ופועל מול ה-database.
3. פעולות נדרשות למימוש באב-טיפוס: לקרוא באמצעות תוכנת ה-client נתונים מהטבלה שנמצאת ב-DB שמותקן ב-server, וכן לכתוב (באמצעות ה-client) נתונים אחרים בטבלה זו.

אופן הפעלת האב-טיפוס:

- יש להכין מראש (ישירות ב-DB, לא ע"י האב-טיפוס) טבלה בשם " Order ", ובה שמורים פרטי המידע הקשורים להזמנת חניה:

parking_space(int FK)	1.
order_number (int PK)	2.
order_date(Date)	3.
confirmation_code(int)	4.
subscriber_id(int FK)	5.
date_of_placing_an_order(Date)	6.

ה- client וה- server יכללו GUI בסיסי ופשוט עם השדות והמרכיבים הנדרשים לביצוע הפעולות הבאות:

- באמצעות ה-client, יש לגשת לפרטים השמורים בטבלה ולהציג את אותם במסך.
- באמצעות ה-client, יש לכתוב ערך חדש (update) בשדות: parking_space , order_date
- באמצעות ה-server, יש להראות את פרטי הקישוריות הרשתית של ה-client המחובר (כתובת IP ושם host), וסטטוס החיבור.
- יש לוודא באמצעות גישה ישירה ל-DB (לא ע"י קריאה באמצעות האב-טיפוס) שהערך החדש שנכתב באמצעות ה-client אכן התעדכן בטבלה.

במימוש האב-טיפוס יש ליישם עקרונות תכנות נכונים של יישום מבוסס Client-Server, הכוללים:

- שימוש נכון במבנה Client-Server בגישה ל-DB.
 - שימוש נכון במנגנון הפצת הודעות.
 - שילוב שאילתות במחלקה הנכונה.
 - שמות מתאימים ליישום הרלבנטי.
- הערה:** הדוגמה שנלמדה בתרגול אינה תבנית ליישום/מימוש – היא רק דוגמה. יש להבטיח בניה של האב-טיפוס בשני קבצי הרצה נפרדים לפי העקרונות ההנדסיים הנכונים.

3. (10%) תוכנית פרויקט מעודכנת (באמצעות תוכנת ToDoList).

- Based on the project planning submitted in assignment 1, present detailed planning of the project schedule on a weekly basis for the period: from the date of submission-1 until the date of submission-2. Planning has to be performed as the planning is done in assignment-1.
- בנוסף יש להגיש בקובץ נפרד את ה-SPRINT BACKLOG עבור הספרינט השני (בין הגשה ראשונה לשנייה). (כלומר יש להגיש את רשימת המשימות שתכננתם לביצוע בשלב הבנייה של הפרויקט: Product Log).

פורמט הגשה:

עליכם להכין קובץ RAR (לחלופין אפשרי: ZIP) בשם: G<GroupNum>_Assignment2 (כאשר <GroupNum> הוא מספר הקבוצה שלכם. לדוגמה, עבור קבוצה 5: G5_Assignment2).

קובץ ה-RAR מכיל את המרכיבים כמתואר להלן:

1. קובץ של Visual Paradigm בשם: G<GroupNum>_Assignment2 שמכיל את כל דיאגרמות ה-UML המבוקשות.
[<GroupNum> הוא מס. הקבוצה שלכם, לדוגמה, עבור קבוצה 5: G5_Assignment2],
2. קובץ ToDoList בשם: G<GroupNum>_ProjectManagement2 (כאשר <GroupNum> הוא מספר הקבוצה שלכם) שמכיל את תכנון הפרויקט המעודכן (פעילויות, לוח זמנים, והקצאת משאבים).
3. קובץ DOC בשם: G<GroupNum>_SprintBacklog שמכיל את ה-SPRINT BACKLOG הנדרש.
- כתבו בראש הקובץ את מספר הקבוצה, שמות כל חברי הקבוצה, מס. ת"ז, כתובות דוא"ל ותאריך ההגשה.
4. 2 קבצי ה-jar של האב-טיפוס בשמות:
G<GroupNum>_Prototype_Client (1) G<GroupNum>_Prototype_Server (2)

לתשומת לב:

יש לשלב את כל קבצי ההגשה **ישירות** ב-ZIP/RAR באופן 'שטוח' 'למעלה' **ללא** יצירת רמת היררכיה נוספת ("תיקיה ראשית") נוספת מעליהם.

יש להקפיד על כל דרישות העריכה וההגשה המפורטות לעיל ושמות הקבצים כפי שהוגדרו כאן.

(*) במקרה של הגשה שאינה תואמת את הנדרש **יופחתו** עד 5 נקודות מציון המטלה (חבל...).

הנחיות הגשה:

יש להגיש את המטלה עד יום ד' 11/05/25 בשעה 8:30.
כל איחור ממועד ההגשה הנדרש יגרור הפחתת נקודות.
ההגשה ל-Assignment2 באתר הקורס ב-MOODLE.
מספיקה הגשה אחת – ע"י אחד.ת מחברי הקבוצה – בשם כל הקבוצה.
יש להקפיד על הנחיות העריכה וההגשה ופורמט ושמות הקבצים כפי שמפורט לעיל.

באחריות הסטודנטים לוודא ביצוע **הגשה סופית** ו**קבלת האישור** על הגשה סופית ממערכת MOODLE (File uploaded successfully).
יש לבדוק שאין אף הגשה של הקבוצה שנשארה במצב **טיוטה**.

במקרה של ספק ניתן לבדוק באותו דף באתר שהקובץ שהעליתם אכן נמצא בו.

מומלץ לבצע את הגשת המטלה זמן מספק מראש **לפני פקיעת המועד** להגשה (לא "בדקה האחרונה"), כדי להימנע מכל מיני בעיות כשלים ו'הפתעות' של הרגע האחרון (stay away from "murphy's law" situation).

בדיקה והערכה:

- בדיקת האב-טיפוס (באמצעות הצגה תפעולית, בדיקת דרישות היישום ובחינת הקוד) תתקיים בנוכחות כל צוות הפרויקט בזמני התירגולים במהלך השבוע של ההגשה.
- מועדי ההגנה לקבוצות השונות יתוזמנו ויתואמו מראש בשבוע שלפני ההגשה (תתפרסם על כך הודעה נפרדת).
- כל חברי הקבוצה חייבים להשתתף בהגנה.
- במקרה של בעיה אובייקטיבית חריגה יש לתאם מראש.
- בהגנה יורצו 2 קבצי ה-Jar (הנפרדים) של האב-טיפוס – Client ו-Server על 2 מחשבים נפרדים.

שאלות לגבי מטלה זו יש לפרסם בפורום Assignment 2 באתר הקורס.

מקורות ידע לתשובות בנושאי המטלה:

1. UML: Activity: קטרינה. Sequence, Class: משה. Package: איליה
2. Project Management: טיראן
3. פעולת המערכת: ענת.
4. Prototype: איליה/קטרינה/טיראן.

בהצלחה