

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 מימוש 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 16.3** של Visual Paradigm) יכיל את הדיאגרמות (סה"כ 60%):
 - **Activity** (16%): תיאור ביצוע התהליכים הבאים באמצעות מערכת התוכנה, כולל הקצאת פעילויות לאובייקטים בטכניקת Swim-Lanes (המלצה: דיאגרמה נפרדת לכל תהליך):
 - תהליך ביצוע מבחן ממבט סטודנט (באופן ידני וגם באופן ממוחשב).
 - יצירת וצפיית דו"חות סטטיסטיות של ראש המחלקה.
 - **Class** (27%): 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** (14%): תיאור מהלך הרצפים הבאים במערכת (המלצה: דיאגרמה לכל רצף):
 - תהליך יצירת מבחן ממבט מרצה.
 - יצירת וצפיית דו"חות סטטיסטיות קשורות להשוואה בין מבחנים שונים של אותו קורס.
 - **Package** (3%): יש לכלול תיאור ארכיטקטורת התוכנה באמצעות Package Diagram.
 - במידת הצורך – השלמה עצמית של לימוד הנושא בעזרת Google: Lucidchart או Visual Paradigm ב-Package diagram guidelines, ולהשתמש בהגדרות והדוגמאות ב-
- ✚ **במטלה 3** תוערך איכות מודל זה מול מימוש ה-Packages בקוד והתיעוד שלהם.

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

2. (20%) **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 "Question", ובה שמורים פרטי המידע הקשורים ל שאלה במבחן:
 1. id (primary key)
 2. subject

- 3. course name
- 4. question text
- 5. question number
- 6. lecturer (author of the question)

- ה- client ו- 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: question text ו- question number.
 - באמצעות ה-server, יש להראות את פרטי הקישוריות הרשתית של ה-client המחובר (כתובת IP ושם host), וסטאטוס 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. (5%) **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 בתחילת הספרינט השני).

4. (15%) עליכם לענות על השאלות הבאות, בהקשר לפעילויות שביצעתם במטלה זו: יש לענות בקצרה ולעניין. כלומר – לענות בדיוק (ובאופן מלא) על השאלות שנשאלו. תשובות כלליות לא יתקבלו.

יש לכלול את השאלות במסמך התשובות.
יש לסמן את כותרות השאלות בבירור.

1. תארו את תהליך הניתוח/תכן ראשוני שביצעתם למרכיב: **בדיקת מבחנים**.
קיום מנחים לתשובה:
- פרטו מה הם הפרטים / השאלות שהתייחסתם אליהם בתהליך הניתוח והחשיבה והתשובות הספציפיות שנתתם להן.
 - התייחסו לקשרים ולמעברים:
1. ממודל תרחישי שימוש Use-case למודל תהליכי מפורט המיוצג בעזרת Activity Diagram,
2. ממודל הפעילויות והתהליכים לתיאור המימוש בתוכנה (Design).
הערה: אין קשר בין פרטי שאלה זו לדרישה הספציפית המופיעה בסעיף 1 במטלה זו.

2. בהרצאה הוגדרה **Reusability** כתכונה של תוצר של תהליך הפיתוח אשר משקפת את היכולת לבצע reuse בהקשר לתוצר זה. בהתאם להגדרה זו, תארו בדיוק (ובהתייחסות ספציפית) ובפירוט איך באות לידי ביטוי 3 הדרישות ליישום מוצלח של Reusability בהקשר של אותם מרכיבים שלא אתם כתבתם או תכנתתם ובחרתם לשלב במערכת שלכם באמצעות Reuse, תוך התייחסות בדוגמאות ספציפיות (לא 'עקרוניות' או 'כלליות') לדרישות הפונקציונליות של המערכת שתכנתתם (ההתייחסות ספציפית בהקשר זה = התייחסות למרכיבים פונקציונליים ספציפיים קונקרטיים (לא גנריים) מתוך התיאור המילולי הראשוני של פעולת המערכת ששאתם מפתחים מהתחלת הסמסטר). לא כולל תהליך Login או זיהוי משתמש. במקרה של OCSF יש להתייחס למרכיבים הפונקציונליים הספציפיים למערכת המפותחת בפרויקט זה. אם יש מי מ-3 הדרישות הנ"ל אשר לא באה לידי ביטוי ב-reuse שביצעתם - הסבירו את הסיבה לכך.

3. א. הערכה כללית:

- מהם היתרונות של מודל UML כעזר לתהליך התכנון?
 - הסבירו איך מתקבלים (מתממשים) היתרונות שציינתם.
 - ציינו דוגמה אחת קונקרטית ממוקדת (לא כללית) (גנרית) ולא Login) מתוך תהליך הניתוח והתכן שאתם בצעתם לשימוש מועיל ב-UML תוך תיאור והתייחסות ספציפית למרכיבים של מערכת "CEMS" שתכנתתם ומידלתם. הערה: אין להסתפק בסופרלטיבים כלליים כמו: "מתאר", "עוזר להבנה", "מועיל", "משפר", וכו'. יש להעמיק ולהסביר את היתרונות, ולתאר בדוגמה ספציפית.
- ציינו קשיים הנובעים מחסרונות של UML שנתקלתם בהם. גם כאן התייחסו ספציפית לתהליך שבצעתם לפיתוח מערכת זו.

ב. ניתוח ודיון:

בהתאם לניסיון שרכשתם במהלך העבודה על מטלה זו, תארו אפשרויות לשינויים ושיפורים במתודולוגית UML אשר נותנים מענה לחסרונות שנתקלתם בהם במהלך ה-design שביצעתם בפרויקט שלכם. הסבירו את תשובתכם תוך תיאור דוגמה ספציפית (לא כללית/גנרית, כולל שמות של רכיבים, לא כולל Login) מתוך עבודתכם.

פורמט ההגשה:

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:

- קובץ של **Visual Paradigm** **בשם**: G<GroupNum>_Assignment2 שמכיל את כל דיאגרמות ה-UML המבוקשות.
- [G5_Assignment2 : עבור קבוצה 5] קובץ **ToDoList** **בשם**: G<GroupNum>_ProjectManagement2 (כאשר <GroupNum> הוא מספר הקבוצה שלכם) שמכיל את תכנון הפרויקט המעודכן (פעילויות, לוח זמנים, והקצאת משאבים).
- קובץ **DOC** **בשם**: G<GroupNum>_SprintBacklog שמכיל את ה-SPRINT BACKLOG הנדרש.
- קובץ **PDF** **בשם**: G<GroupNum>_Answers.Ass.2 שמכיל את התשובות לשאלות. **כתבו בראש הקובץ** את **מספר הקבוצה**, **שמות** כל חברי הקבוצה, **מס. ת"ז**, **כתובות דוא"ל** ו**תאריך ההגשה**.
- 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 "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 ג' 16.5.23 שעה 14:30 .
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 וקבלת 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 :ענת. Package : איליה
2. Project Management : טיראן
3. שאלות: אבי.
4. Prototype : איליה/קטרינה/טיראן.

בהצלחה