

מטלה סופית: מימוש המערכת

כללי

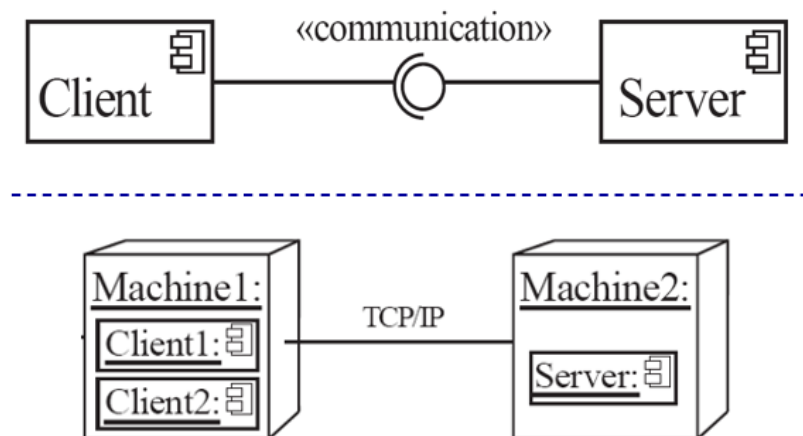
עליכם לבנות ולהגיש מערכת תוכנה פועלת היטב ובאופן מלא. המערכת תממש את הדרישות הפונקציונליות והלא-פונקציונליות למערכת "CEMS" שתוארו במסמך בעל הכותרת "תיאור פרויקט" שנמסר בהתחלת הסמסטר.
ראו מידע קריטי לגבי הנחיות הגשה והבהרות חשובות בסוף המסמך.

הגדרות המטלה

פורמט: כל המלל הכתוב במטלה (שמות, תיעוד, וכו' למעט מענה לשאלות בסעיף 3) יהיה **באנגלית**. המטלה הסופית של הפרויקט כוללת 5 מרכיבים (שלושה מהם אפקטיביים בקורס זה):

1. (73%) מימוש

- עליכם לבנות מערכת תוכנה המממשת את דרישות הפרויקט באמצעות קוד Java.
- מימוש ארכיטקטורה מבוססת-TCP/IP, כולל ריצה עצמאית (ע"י jar) על שני מחשבים נפרדים – כפי שמתואר בדיאגרמות Component + Deployment להלן:



- Database רלציוני שמהווה DBMS מלא.
- העברת מידע ו/או קבצים (בהתאם לדרישות) השמורים בסביבת השרת (server) והצגתם בסביבת הלקוח (client).
- יש לכלול במערכת ממשק משתמש גרפי לתפעול פרמטרי התקשורת בין Client-Server.
- נדרש שימוש בקריטריוני איכות למבנה, תכן ופועלת התוכנה וממשק המשתמש כפי שנלמדו בהרצאות הקורס.

2. (10%) מבנה ותיעוד

- עליכם להכין את תיעוד קוד המערכת בעזרת מנגנון Javadoc. התיעוד כולל את תיאור מבנה התוכנה ומרכיבי ממשק המשתמש. יש לתאר את המחלקות וכל מרכיביהן.
- מרכיב המבנה בציון סעיף זה מתייחס לחלוקת הקוד ל-Packages בהתאם למודל שהוגש במטלה 2:

- יש לציין במפורש במלל (text) האם נוספו שינויי מבנה בקוד ביחס למודל Package Diagram שהוגש במטלה 2 או לא. אם יש שינויים, יש להצביע עליהם ולתאר במלל מהם השינויים - תוך התייחסות פרטנית לשמות ה-packages וה-class שמומשו בקוד.

3. (17%) עליכם לענות בקצרה ולעניין על השאלות הבאות:
(יש לענות **בדיוק** ובאופן מלא על השאלות שנשאלו כאן)

- ❖ יש לכלול את השאלות במסמך התשובות.
- ❖ יש לסמן את כותרות השאלות בצבע (באופן שהן מסומנות כאן).
- ❖ יש להקפיד לענות על כל מרכיבי השאלות.

✚ **בראש מסמך התשובות** יש לכלול **תמונה** של מסך הכניסה (הראשי) למערכת.
- לא מסך החיבור הראשוני בין Client-Server אלא המסך של המערכת עצמה.
- אין צורך במסך של ה-Server.

שאלה 1

תארו את תהליך התכנון (design) שביצעתם עבור המערכת "CEMS" באופן הבא:
(א) - תיאור תהליך התכנון עבור היכולת: יצירת מבחנים.
- תארו דילמות הנדסיות ספציפיות שעסקתם בהן בתכנון של יכולת זו ואופן פתרון.
לצורך זה השתמשו בפורמט של תיאור Design issue כפי שמופיע בהרצאה על Design.
תארו את הדילמות, השיקולים וקבלת ההחלטות שלכם והסבירו את הפתרונות שבחרתם.
(ב) ציינו אילו מהעקרונות (*) שנלמדו בהרצאות הקורס בנושאים: Design, Architecture, Reuse, Design patterns ו- Design patterns שימשו אתכם בתהליך התכנון הכלל-מערכתי שביצעתם והסבירו באיזה אופן הבאתם אותם לידי ביטוי – באמצעות **דוגמאות קונקרטיות ספציפיות** מתוך המערכת "CEMS" שפיתחתם (לא לכלול תיאורים כלליים-גנריים כמו: timer, JAVA packages, OCSF - באופן כללי, וכו'. יש לפרט בהקשר קונקרטי מפורט של המערכת).
(*) לכל אחד מהעקרונות האלה, הסבירו מה הם היתרונות המתקבלים משימוש בהם.

שאלה 2

תארו את תהליך המימוש (implementation) שביצעתם עבור המערכת "CEMS" בהקשרים:
(א) **בדיקות**. תארו את תהליכי הבדיקות השונים שבצעתם במהלך פיתוח הפרויקט שלכם.
■ ציינו את מאפייני תהליכי הבדיקות שביצעתם תוך התייחסות לעקרונות שנלמדו בהרצאות בנושאי בדיקות תוכנה, ותוך מתן **דוגמאות ספציפיות** (לא כללית/גנרית ולא על Login) שביצעתם (או לא ביצעתם) במהלך הפרויקט ע"י תיאור מפורט של בדיקות מרכיבים **ספציפיים** של מערכת "CEMS" ("ספציפי" כלומר: לא בהתייחסות כוללת גנרית כמו: "שדות ריקים", "התחברות לשרת", "כתיבה ל-DB", "דוחות" וכו'. מה כן: יש לפרט מה בדיוק נבדק בדגש על תהליכים ספציפיים של המערכת המפותחת).
(ב) **אינטגרציה**. תארו את תהליכי האינטגרציה שביצעתם במהלך תהליך הפיתוח.
■ תארו את מהלך האינטגרציה – ציינו לוחות זמנים ופרוצדורה (תהליך) - באמצעות **דוגמאות קונקרטיות ספציפיות** (לא כלליות) מתוך המערכת "CEMS" שפיתחתם.
■ ציינו באילו כלי תוכנה השתמשתם באינטגרציה והסבירו באיזה אופן.
■ ציינו אילו מהעקרונות שנלמדו בהרצאה בהקשר של אינטגרציה יושמו בתהליך האינטגרציה שביצעתם, והסבירו באיזה אופן.

שאלה 3

תחקור והפקת לקחים: התייחסו לאופן שבו התנהלתם לגבי 2 מרכיבים של ביצוע הפרויקט:
(א) **תיאום פעילויות ושיתוף** בין חברי הצוות בפיתוח כולל גישה לניהול גרסאות:
תארו את השיטה שלפיה פעלתם בהקשרים אלה בשלב מימוש התוכנה, וציינו יתרונות וחסרונות שלה.
יש להתייחס גם לתהליך העבודה - לא להתמקד רק בכלים ואספקטים טכניים.
(ב) **ניהול הפרויקט בשלב הבניה (Construction)** – קידוד, שילובי קוד (אינטגרציה מערכתית) ובדיקות.
ציינו באופן **פרטני**, בהתייחסות **ספציפית** לפיתוח המערכת "CEMS", איך פעלתם בשלב זה של הפיתוח (כולל: תיאור התנהלות התהליך ההנדסי (לא דינמיקה ועבודת הצוות), בקרה: איך טיפלתם בבעיות טכניות הקשורות לתהליך הפיתוח, וכו').
אם היו קשיים מה הסיבה לכך? מה הייתם משנים בדיעבד בגישתכם למרכיב זה של תהליך הפיתוח מבחינת האספקטים הרלבנטיים של הנדסת תוכנה?

4. (0%) בדיקות.

ביצוע בדיקות לפרויקט ייבדק ויוערך במסגרת הקורס הצמוד "מבוא לבדיקות תוכנה".
הציון יהווה חלק מציון הקורס הנ"ל (הציון לבדיקות לא נכלל בציון של קורס שיטות).

5. (0%) ממשק משתמש.

מרכיב ממשק המשתמש של הפרויקט ייבדק ויוערך במסגרת הקורס הצמוד "ממשק אדם-מחשב". הציון יהווה חלק מציון הקורס הנ"ל (ולא נכלל בציון של קורס שיטות).

תובנות מועילות לשלב הביצוע (Implementation)

להלן תובנות שנכתבו בעבר ע"י סטודנטים כתשובה לשאלה 3 ('תחקור והפקת לקחים'):

1. כדאי להקדיש תשומת לב לכתיבת קוד באופן יעיל ופשוט. היו לנו מקרים שחבר צוות כתב אלגוריתם שגם חבר צוות אחר היה צריך לכתוב. ובגלל שהדברים לא תמיד נכתבו בצורה ברורה זה גרם לשכפול קוד כי שני חברי צוות בעצם כתבו את אותה פונקציה. לדוגמה
2. כדאי לכתוב את התיעוד במהלך הכתיבה של הקוד. כאשר מטפלים בתיעוד רק לקראת סוף הפרויקט – זה לא יעיל ולוקח המון זמן. כן – בסוף תיעדנו אבל לא בצורה שנדרשת על מנת להפיק JavaDoc טוב, ואם היינו מתעדים מיד במהלך העבודה היינו חוסכים זמן לעצמנו.

מומלץ ללמוד מניסיונם של אחרים.

פורמט הגשה

❖ **עבור מרכיבים 1,2,3 עליכם להכין קובץ RAR בשם:** `G<GroupNum>_Assignment3`
[כאשר `<GroupNum>` – מס. הקבוצה שלכם, לדוגמה (לקבוצה 5): `G5_Assignment3`]
המכיל **4 תיקיות** ("ספריות") **בשמות** כמצוין כלהלן ב: 1. – 4, ובהן נמצאים הקבצים כמפורט כאן:

1. **PROJECT**: קובץ הפרויקט של Eclipse שכולל את כל הקוד שקיים במערכת שלכם.
שם הקובץ: `G<GroupNum>_Assignment3-Project` [`<GroupNum>` – מס. הקבוצה].
2. **JAR**: 2 קבצי ה-`jar` שמאפשרים את הרצת המערכת, **בשמות:** `G<GroupNum>_client` ו- `G<GroupNum>_server`, קובץ ה-`database`, וקבצים נוספים במידת הצורך.
3. **ANS**: קובץ **PDF** **בשם:** `G<GroupNum>_Answers.Ass.3` שמכיל את התשובות לשאלות.
יש לכתוב **בראש הקובץ** את **מספר הקבוצה**, **שמות** חברי הקבוצה, **כתובות דוא"ל ותאריך ההגשה**.
4. **DOC**: קובץ **RAR** **בשם:** `G<GroupNum>_Assignment3-Doc` שמכיל
(1) את תיעוד הפרויקט (באמצעות Javadoc).
(2) תיעוד מילולי המתייחס ל**מבנה** הקוד לפי המוגדר בסעיף 2. 'מבנה ותיעוד'.

לתשומת לב:

יש לשלב את 4 תיקיות ההגשה **ישירות** בקובץ ZIP/RAR באופן '**שטוח**' "ישר למעלה" - **ללא** 'יצירת תיקיה ראשית' נוספת מעליהם.

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

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

❖ הנחיות להגשה עבור **מרכיבים 4,5** יינתנו במסגרת הקורסים הרלבנטיים.

הנחיות הגשה:

יש להגיש את מרכיבים 1, 2, 3 של המטלה על פי הפורמט המתואר לעיל עד יום ג' 20.6.23 שעה 14:30. כל איחור ממועד ההגשה הנדרש יגרור הפחתת נקודות. ההגשה ל- Assignment3 באתר הקורס ב-MOODLE. כל הקבוצות מגישות בתאריך הנקוב לעיל. לא תתאפשר הגשה מאוחרת.

באחריות הסטודנטים לוודא ביצוע הגשה סופית וקבלת האישור על הגשה סופית ממערכת MOODLE (יש ללחוץ על "שליחת הגשה למתן ציון"). יש לבדוק שאין אף הגשה של הקבוצה שנשארה במצב טיוטה. במקרה של ספק ניתן לבדוק באותו דף באתר שהקובץ שהעליתם אכן נמצא בו.

מומלץ להקדים ולבצע את הגשת המטלה זמן מספק מראש לפני פקיעת המועד להגשה (לא "בדקה האחרונה"), כדי להימנע מכל מיני בעיות כשלים והפתעות של הרגע האחרון.

הערה: כפי שנלמד בקורס, השלב האחרון בתהליך הפיתוח עוסק בפעילות של Phase 4: **Transition** - הכוללת: אריזה, התקנה והפעלה. ביצוע שלב זה מהווה מרכיב של דרישות הפרויקט. לפיכך, גישה מוטעית היא ליצור את גרסת ההגשה של הפרויקט "ברגע האחרון" - כי אז אם משהו משתבש (למשל הפרויקט לא רץ טוב בסביבת הבדיקה או "לא עולה") אין זמן לפעולות לתיקון – ויופחתו נקודות מהציון. יש לבדוק את פעולת הפרויקט לפני ההגשה בגרסת ההגשה בסביבת הבדיקה – בשני מחשבים נפרדים במכללה.

בדיקת הפרויקט

בדיקת מרכיבים 1, 4, 5 של המטלה תתבצע באמצעות הפעלה והצגה של המערכת בשלושה חלקים:

1. בדיקה של מרכיב מס. 1 של המטלה (**מימוש**):
בדיקת הפרויקט תתקיים בנוכחות כל צוות הפרויקט בזמני התירגול במהלך השבוע של ההגשה. הבדיקה של מטלה 3 היא חד-פעמית (אין מועד ב').
כל חברי הקבוצה חייבים להשתתף בבדיקה. במקרה של בעיה אובייקטיבית חריגה יש לתאם מראש.
זמן ומיקום מדויק לביצוע הבדיקה יתואם לכל קבוצה מראש.
הנחיות לתיאום ורישום לבדיקה יפורסמו ב-MOODLE.

2. בדיקה של מרכיב מס. 4 של המטלה (**בדיקות**):
בהתאם להנחיות שימסרו בקורס: "מבוא לבדיקות תוכנה". זמן מדויק יתואם לכל קבוצה לקראת הבדיקה במסגרת הקורס הנ"ל. הציון של בדיקה זו אינו נכלל בציון של קורס 'שיטות'.

3. בדיקה של מרכיב מס. 5 של המטלה (ממשק המשתמש):
בהתאם להנחיות שימסרו בקורס: "ממשק אדם-מחשב". זמן מדויק יתואם לכל קבוצה לקראת הבדיקה במסגרת הקורס הנ"ל. הציון של בדיקה זו אינו נכלל בציון של קורס 'שיטות'.

הבהרות חשובות:

- במהלך כל תהליך הפיתוח עליכם לדאוג לגיבוי נאות של עבודתכם באופן שיאפשר התאוששות מהירה ומוצלחת מ'קטסטרופות'.
בפרט עליכם להכין את עצמכם מראש להתאוששות ממצבים כמו למשל: "הייתה הפסקת חשמל והכל נמחק לנו", "הדיסק במחשב קרס", "גיט מחק לנו הכל" או כל תרחיש הרסני אחר.
- המערכת תיבדק כפי שהוגשה בקבצי Jar – Client – Server.
לא יתאפשר עדכון תוכנה 'ברגע האחרון'.
- יש להכין מראש database שכולל נתונים שיאפשרו את ביצוע הבדיקה, ולהתקין אותו לפני הפעלת המערכת לבדיקה. יש להכין (בפורמט טקסט) ולהציג בבדיקה תיאור של תכולת ה-DB שהוכן מראש לבדיקה (כולל טבלאות, נתונים).
- בבדיקה יורצו 2 קבצי Jar נפרדים (Client-Server).

שאלות לגבי מטלה זו יש לפרסם בפורום Assignment 3 באתר הקורס.

מקורות ידע לתשובות בנושאי המטלה:

- עיבוד מידע / דו"חות: קטרינה.
 - תיעוד (Javadoc), מבנה (Package): איליה.
 - שאלות: אבי.
 - הגשה, התקנה: טיראן.
- הפורום ייסגר כ-48 שעות לפני מועד הגשת המטלה (כך שיש לסיים את כל השאלות והבירורים לפני מועד זה).
אין לצפות לקבלת תשובות מצוות הקורס במהלך סוף השבוע לפני ההגשה. פניות ישירות במייל יענו במידת האפשר רק בענייני חירום. לא יענו שאלות הבהרה/חידוד/ויישוב בקשר לתכולת הפרויקט.

בהצלחה