

מסמך מלווה לפרויקט תכנותי: שרת משחק

פרויקט במסגרת הקורס:

סדנא בתכנות מתקדם בשפת JAVA (20503)

**מגיש: גבריאל לדזרטי**

**ת.ז: 306581232**

**מוגש לידי: גב' תמר בניה**

**המחלקה למדעי המחשב**

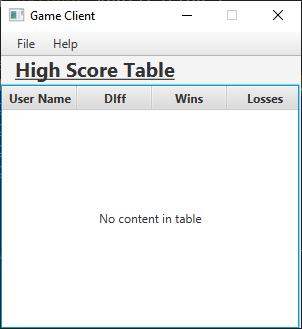
**מאי 2021**

**מדריך למשתמש**

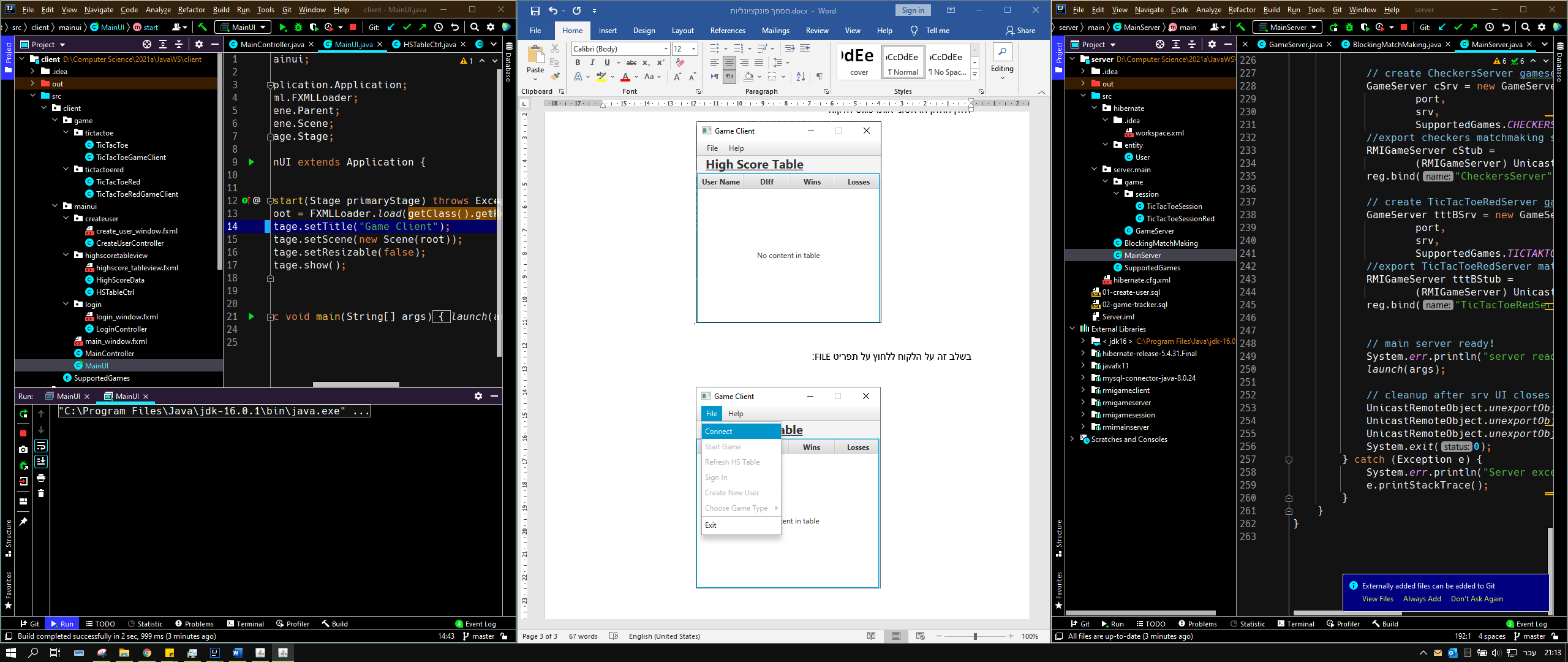
פרויקט זה מממש שרת משחק המצוות שחקנים שונים על פי סוג המשחק אותם הם בוחרים.

בעת פתיחת אפליקציית צד הלקוח:

להלן החלון הראשוני אותו פוגש הלקוח –

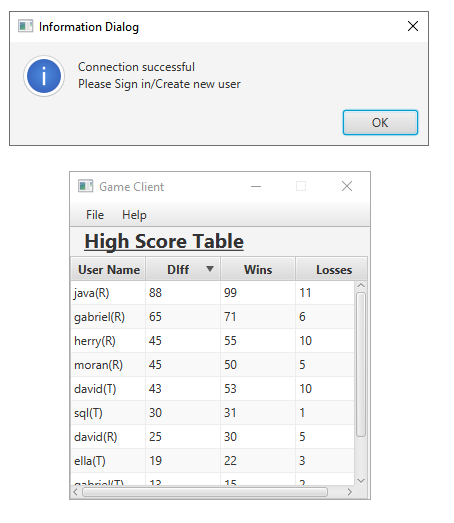
.

בשלב זה על הלקוח ללחוץ על תפריט FILE:

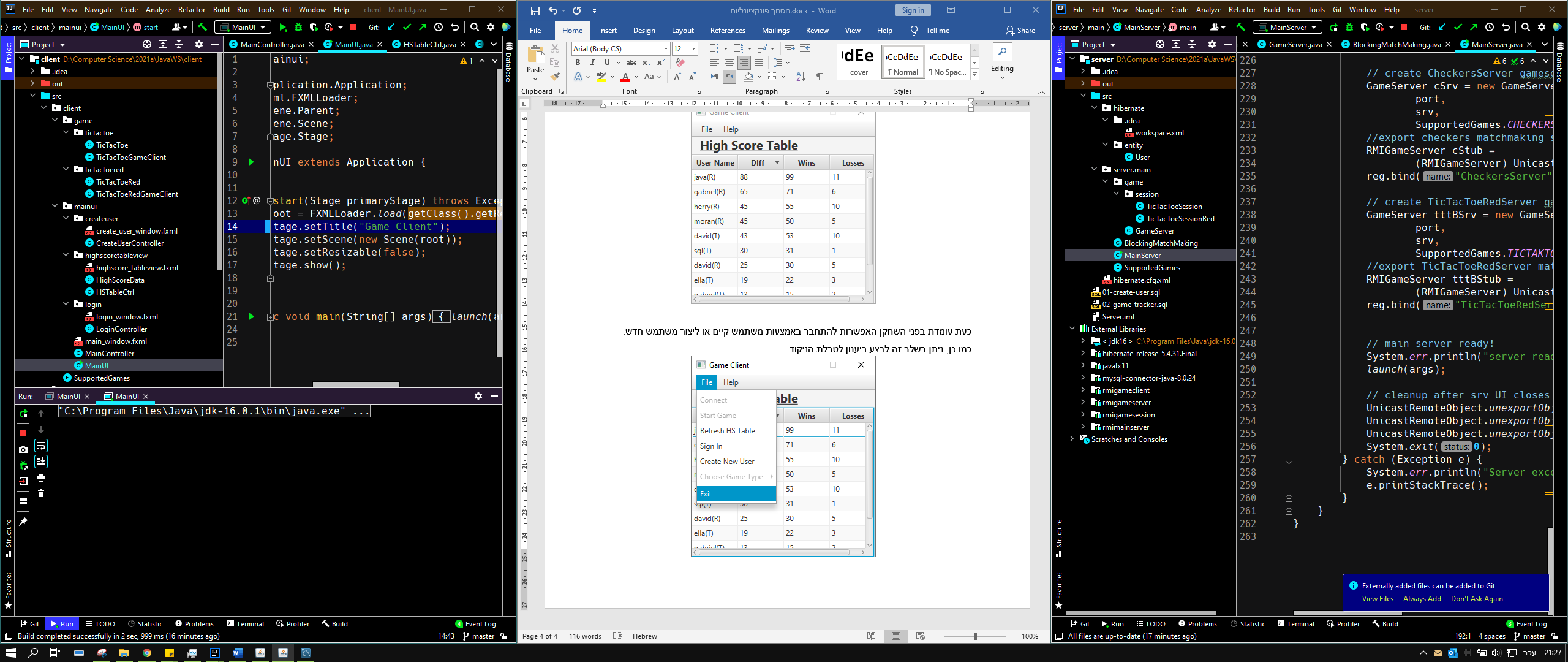


כעת על יש ללחוץ על כפתור הconnect על מנת להתחבר לשרת הראשי.

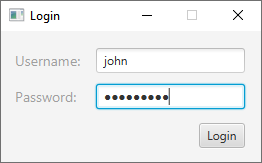
לאחר החיבור לשרת הראשי, מוצג על גבי החלון הראשי לוח תוצאות גבוהות של המשחקים השונים:

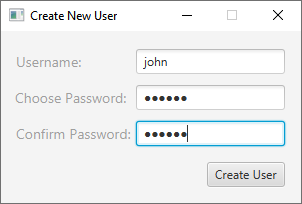


כעת עומדת בפני השחקן האפשרות להתחבר באמצעות משתמש קיים או ליצור משתמש חדש. כמו כן, ניתן בשלב זה לבצע ריענון לטבלת הניקוד.

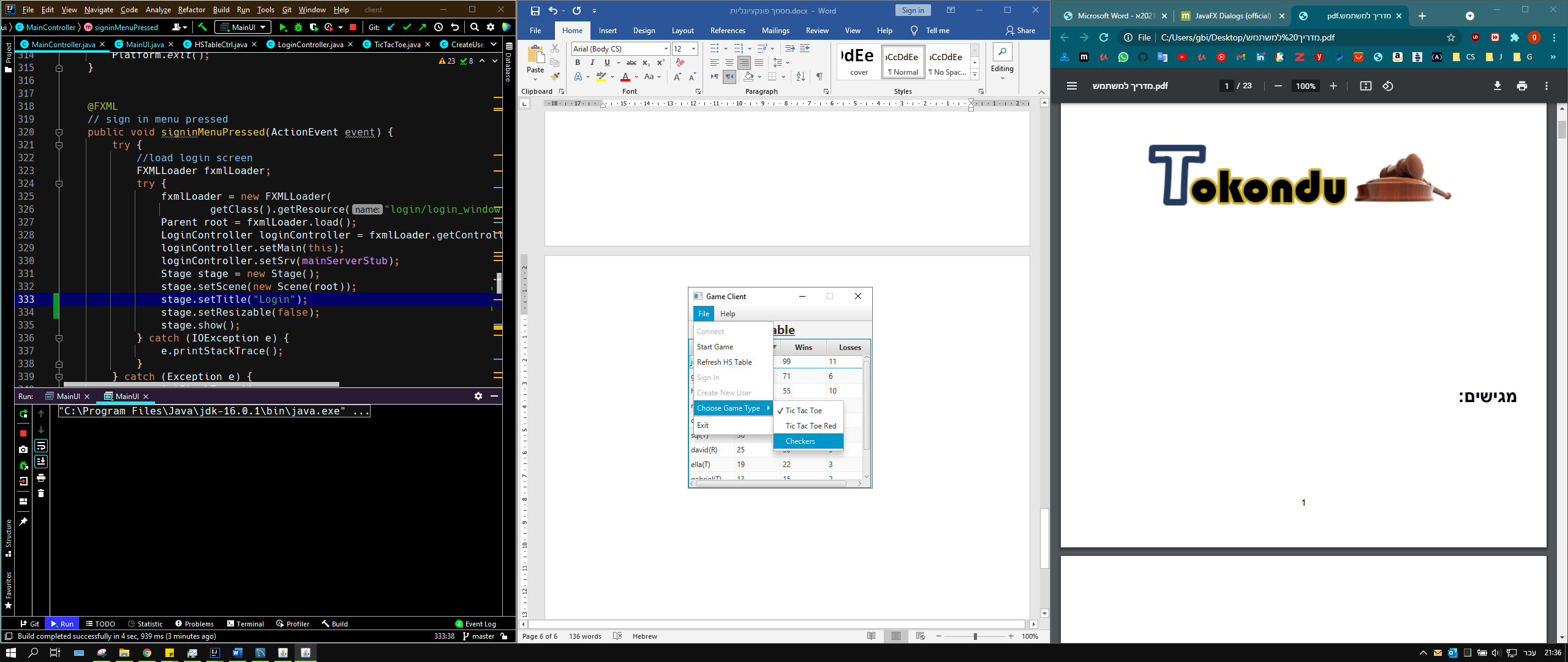


להלן מסכי ההתחברות/יצירת משתמש חדש:



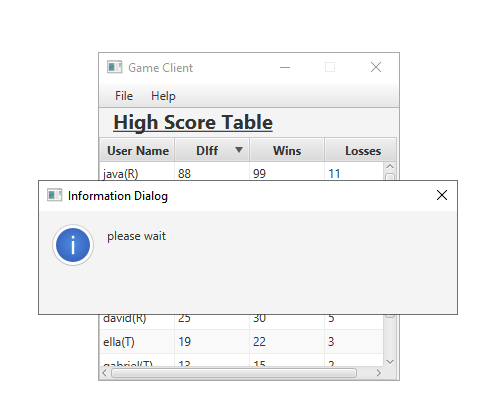


לאחר ההתחברות, על השחקן לבחור את המשחק אותו הוא מעוניין לשחק מתוך המשחקים הזמינים בשרת:

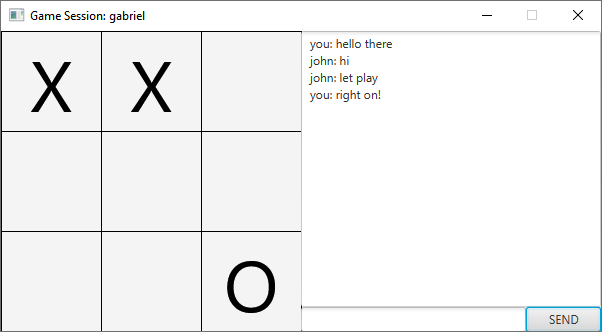


לאחר בחירת המשחק, יש ללחוץ על לחצן התחלת המשחק תחת תפריט File.

לאחר הלחיצה, ישנו מסך המתנה לתחילת המשחק, זאת כאשר יימצא יריב.

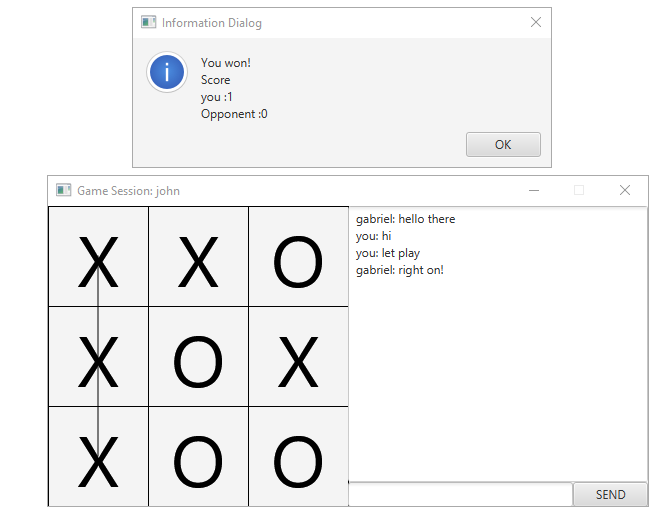


לאחר שנמצא יריב, מתחיל המשחק:



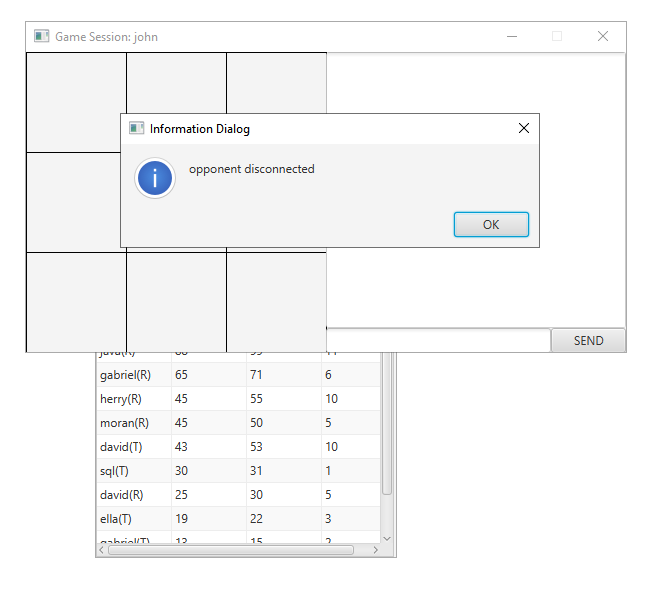
מימין ללוח המשחק קיים חלון המאפשר צ'אט עם היריב.

לאחר ניצחון/תיקו, מופיע חלון קופץ המכיל את הניקוד העדכני בין השחקנים הנוכחיים. לחיצה על OK או סגירת החלון תתחיל משחק חדש בניהם. וזאת אך ורק במידה ושני השחקנים חפצים בכך.



סגירת חלון המשחק הראשי יסיים את המשחק.

הודעה על יציאת היריב תופיע אצל השחקן השני. אישור החלון תחזיר את השחק לחלון הראשי.



**תיאור פונקציונלי**

**טכנולוגיות בשימוש המערכת**

hibernate-release-5.4.31.Final –

פריימוורק ORM Object–relational mapping – מיפוי אובייקטים רציונלי. המממש את JPA ומשתמש ב-JDBC בכדי לאפשר תקשורת עם בסיסי נתונים רציונליים כגון mysql בפרויקט זה.

javafx11 –

חבילה כוללת ליצירת ממשק משתמש מסוגים שונים כגון ממשק למכשירים ניידים, desktop וכד'.

mysql-web-community-8.0.24.0 –

מסד נתונים יחסי ורב משתמשים בשפת SQL. בבעלות ORCALE

mysql-connector-java-8.0.24 –

דרייבר JDBC עבור mysql.

IntelliJ IDEA 2021.1.1 (Ultimate Edition) -

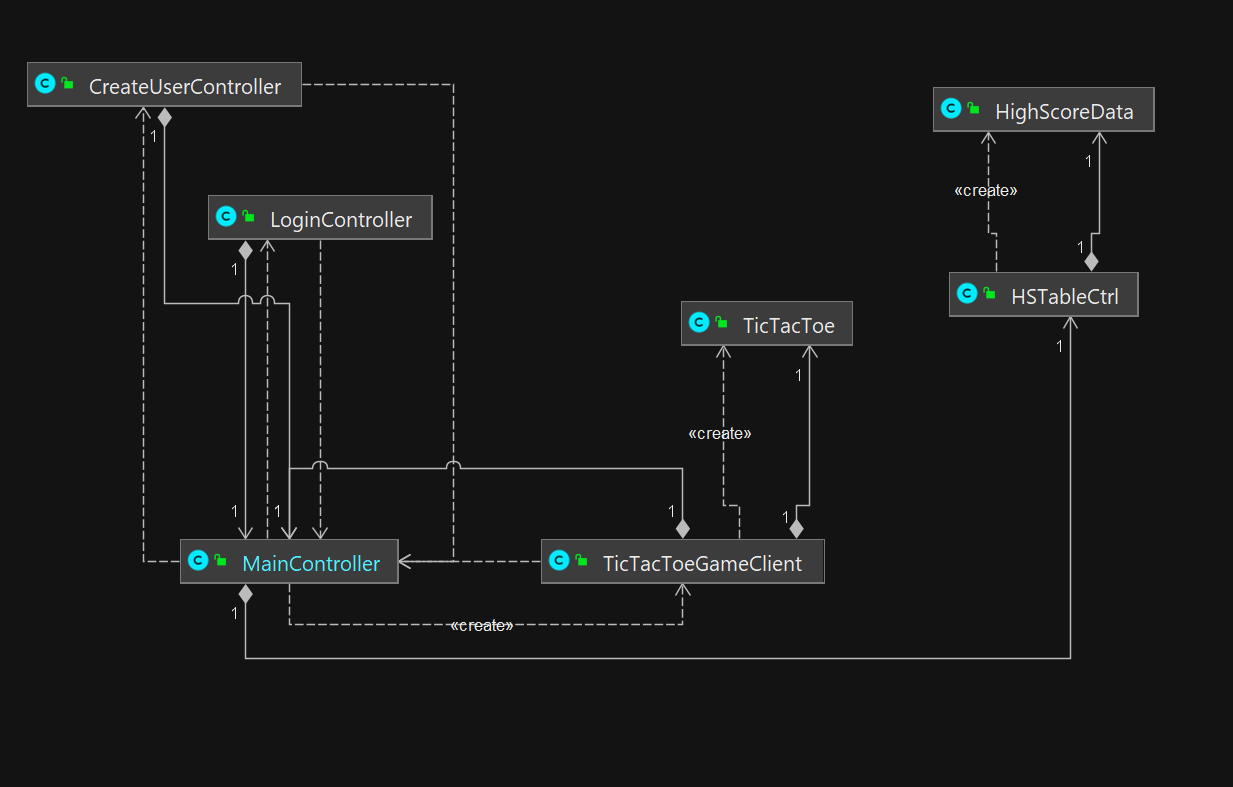
IDE עבור java מבית jetbrains.

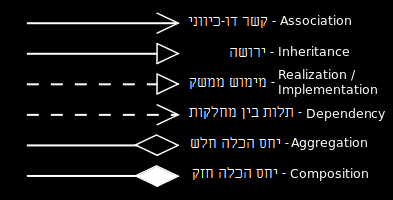
**מבנה המערכת**

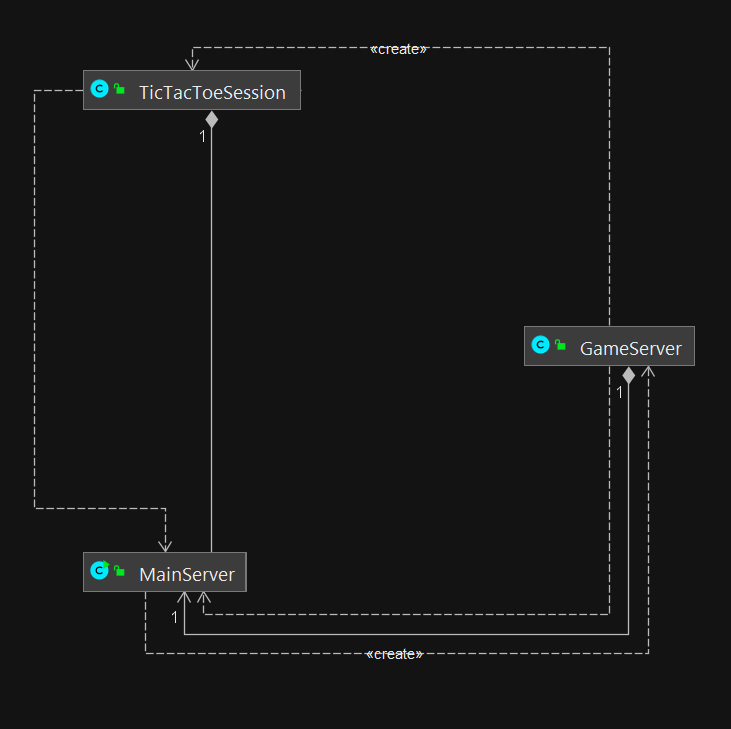
**קבצי קונפיגורציה:**

Hibernate.cfg.xml – קובץ קונפיגורציה עבור חיבור למסד הנתונים. מכילה הגדרות הכרחיות כגון: כתובת בסיס הנתונים, שם משתמש וסיסמא עבור מנהל בסיס הנתונים. הגדרות דיאלקט וכדומה.

**תיאור קשרים בין מחלקות המערכת**

**צד לקוח**

****

**צד שרת**

**תיאור כללי של התקשרות שרת - לקוח**

מתודת ה-main במחלקה MainServer מורצת ומאתחלת שרתי ציוות שחקנים, שרת עבור כל משחק נתמך. שרתים אלו נרשמים בregistry ומיישמים את ממשק RMIGameServer.

הלקוח מצידו מתחבר תחילה לשרת הראשי – MainServer ולאחר שקובע את סוג המשחק הדרוש, נשלחת בקשה לשרת שבתורו שולח חזרה את פרטי ההתקשרות לשרת הציוות הרלוונטי.

לאחר שבשרת הציוות מתחברים מספר שחקנים הדרוש עבור קיום משחק, נפתח מופע של RMIGameSession ופרטיו נשלחים ללקוחות הללו. אובייקט מרוחק זה מנהל את המשחק הפעיל בין השחקנים שבשלב זה מקבלים את פרטי ההתקשרו עמו.

כלל לוגיקת המשחק נמצאת בצד השרת ובאובייקט RMIGameSession בפרט.

**מחלקות צד שרת**

**חבילת server.main**

MainServer.java – מחלקה ראשית המנהלת את כל התקשורת עם בסיס הנתונים. בנוסף מאתחלת מבני נתונים לניהול ציוות שחקנים השונים באפשרויות המשחק השונות.

מחלקה זו מממשת ממשק מרוחק – RMIMainServer – ממשק זה מגדיר מתודות המשמשות לתקשורת עם מסד הנתונים (CRUD) ושאילתות כלליות.

ממשק זה מגדיר גם מתודות לחיבור וניתוק של לקוחות לשרת.

BlockingMatchMaking.java - מחלקה המממשת מבנה נתונים לציוות שחקנים. בלב המחלקה תור, הproducer הינם המשתמשים המעוניינים להתחבר ולשחק. הconsumer הינו שרת המממש את ממשק הRMIGameServer (יוצג בהמשך). ממומש בwait notify.

SupportedGames.java – Enum המכיל את המשחקים הנתמכים בשרת המשחק.

**חבילת server.main.game**

GameServer.java – מחלקה המממשת את הממשק המרוחק RMIGameServer. מחלקה זהו אחראי על לוגיקת הmatchmaking.

מחלקה זו מאתחלת בעת יצירתה תהליך המשמש לציוות שחקנים שונים. לקוח המעוניין בכך משתמש במתודות הממשק המרוחק בכדי להיכנס למבנה הנתונים האחראי על הציוות. למעשה התהליך המורץ בעת ביצוע המתודה המרוחק connect מהווה תהליך producer. התהליך המאותחל על ידי המחלקה הינו הconsumer. ציוות מתאפשר כאשר מספר השחקנים הדרוש נכנס למבנה הנתונים.

ממשק RMIGameServer מכילה 3 מתודות – connect, disconnect וgetSupportedGames. שלושת המתודות לשימוש שחקנים מרוחקים.

**חבילת server.main.game.session**

TicTacToeSession.java – מחלקה זו מנהלת את מהלך המשחק בין השחקנים. לוגיקת המשחק נמצאת במחלקה זו שבשרת.

מחלקה זו מממשת את הממשק המרוחק RMIGameSession, ממשק זה מגדיר בין השאר את מתודת הmove אשר באמצעותה מנסה שחקן מרוחק לדווח על מהלך שביצע. המהלך נבדק אל מול לוגיקת המשחק הממומש במחלקה ותשובה בוליאנית מוחזרת לשחקן המרוחק. בנוסף ממשק זה מגדיר את אופן הצ'אט בין השחקנים.

**חבילת hibernate.entity**

User.java – מחלקה המייצגת את המידע הנשמר בבסיס הנתונים. מחלקה זו באה בשימוש על ידי ספריית הhibernate בעת תקשורת עם בסיס הנתונים.

**מחלקות צד לקוח**

**מחלקת ui**

AlertBox **–** מחלקה זו יורשת מAlert מתוך Javafx. כל תכליתה הוא יצירתdialog messages.

**חבילת hibernate.entity**

User.java – מחלקה המייצגת את המידע הנשמר בבסיס הנתונים. צד הלקוח משתמש במחלקה זו כדי לפרש מידע המגיע מהשרת ומתוך בסיס הנתונים.

**חבילת client.mainui**

MainUI.java – מחלקה הטוענת את קובץ הFXML הראשי המייצג את החלון המשתמש הראשי של צד הלקוח.

MainController.java – מחלקת השליטה הראשית. מחלקה זו מנהלת את התקשורת עם השרת הראשי ובפרט הזדהות הלקוח/יצירת לקוח חדש, התחלת משחק חדש ובנוסף פתיחת חלונות חדשים – לוח המשחק, התחברות ויצירת לקוח חדש.

main\_window.fxml – קובץ הגדרת חלון הממשק המשתמש הראשי javafx.

**חבילת client.mainui.login**

LoginController.java – מחלקת שליטה בחלון הזדהות המשתמש.

login\_window.fxml – קובץ הגדרת חלון ההזדהות.

**חבילת client.mainui.highscoretableview**

HighScoreData.java – מחלקה המגדירה אובייקט מידע לשימוש עבור תצוגת לוח ניקוד מסוג tableview מספריית javafx.

HSTableCtrl.java – מחלקה היורשת מtableview. מגדירה את אופן התצוגה של לוח התוצאות במסך הממשק הראשי.

highscore\_tableview.fxml – קובץ הגדרת תצוגת לוח תוצאות גבוהות.

**חבילת client.mainui.createuser**

CreateUserController.java – מחלקת שליטה בחלון יצירת משתמש חדש.

create\_user\_window.fxml - קובץ הגדרת חלון יצירת משתמש חדש.

**חבילת client.game.tictactoe**

TicTacToe.java – מחלקה האחראית על יצירת ממשק המשתמש בעת משחק פעיל. מתקשרת עם GameSession עבור עדכון מהלך שחקן.

TicTacToeGameClient.java – מחלקה המממשת את Runnable וממשק מרוחק RMIGameClient. מחלקה זו אחראית על יצירת ממשק המשתמש עבור משחק פעיל.

ממשק RMIGameClient – מגדיר מתודות לשימוש על ידי gameserver לעדכון צד הלקוח עם פרטי ההתקשרות לשרת המשחק – gamesession. כמו כן מכיל מתודות לשימוש gamesession בעת ניהול המשחק ובפרט עבור צ'אט המשחק.

**הוראות התקנה והרצה**

1. יש להוריד את ספרית hibernate:

<https://hibernate.org/orm/releases/5.4/>

1. לאחר מכן את יש להתקין את mysql גרסת web community:

<https://dev.mysql.com/downloads/installer/>

1. להוריד ולהתקין mysql connector for java – ממשק JDBC עבור בסיס הנתונים

<https://dev.mysql.com/downloads/connector/j/>

1. יש לפתוח שרת חדש במחשב, בכדי לאתחל שרת עבור הפרויקט יש להריץ את קובץ הסקריפט 02-game-tracker.sql.

את שם השרת יש להגדיר כ- (נעשה בסקריפט):

game\_server\_db\_tracker

משתמש מסד נתונים:

gameserver\_db

סיסמא:

Admin

1. יש להוריד javafx:

<https://gluonhq.com/products/javafx/>

1. יש לחלץ את כלל התיקיות מקובץ proj.rar.
2. כדי לפתוח את צד הלקוח יש לפתוח את תיקיית client כפרויקט בIntellij
3. כדי לפתוח את צד הלקוח יש לפתוח את תיקיית server כפרויקט בIntellij
4. מתודות הmain נמצאות במחלקות MainServer ו- MainUI עבור השרת וצד הלקוח בהתאמה.
5. יש להוסיף את כל קבצי הJar כספריות חיצוניות בפרויקט:

בintellij:

File->project structure->libraries->add->

ולבחור את כל קבצי הjar בתיקיות הבאות:

\hibernate-release-5.4.31.Final\required

\javafx-sdk-15.0.1\lib

\mysql-connector-java-8.0.24

1. כעת ניתן להריץ את ממשק המשתמש והשרת.