

Software Documentation

Database and Schemes:



הסבר כללי:

מסד הנתונים מורכב בעיקרו מ 4 טבלאות: טבלה למטופל, טבלה למשחק, טבלה לצורות וטבלה למרקמים. ישנה עוד טבלה אחת שתפקידה לקשר בין מטופלים למשחקים שהם שיחקו. בטבלה של המטופלים נשמר כל המידע האישי של המטופל, כמו תעודת זהות, שם, מין וכד'. בטבלה של המשחקים נשמר כל המידע על טיב המשחק כגון תעודת זהות ייחודית לכל משחק, סוג המשחק, הגבלת הזמן באותו משחק וכד'. הטבלה המקשרת מכילה תעודות זהות של משתמשים ושל משחקים. תעודת זהות של משתמש שנמצאת ביחד עם תעודת זהות כלשהי של משחק, מעידה על כך שהמטופל הזה שיחק את המשחק הספציפי שזו התז. הייחודית שלו. שתי הטבלאות הנוספות מכילות את תוצאות זמן התגובה של המטופלים במשחקים לפי סוג המשחק. (במשחק משולב יופיעו תוצאות בשתי הטבלאות). הקישור למשתמש ספציפי נעשה דרך תעודת זהות של אותו משתמש.

תיאור הסכמות

patient: This table describes a patient.

values:

- patient_id – the unique id of the patient. Used as identifier.
- patient_name – the name of the patient.
- patient_gender – the gender of the patient.
- dominant_hand – a char that represents the dominant hand of the patient.
- birth_date – the birth date of the patient.
- patient_type – the type of the user (patient or tester) – for research purpose.

game: This table describes a game.

values:

- game_id – the id of the game.
- game_type – the type of the game (shapes\textures\both).
- num_recognized_buttons – the number buttons the patient recognized in the game.
- game_date – the date when the game was played.
- time_limit – the time limit of the current game.
- dominant_hand – an indicator if the patient used his/her dominant hand or not.

patient_game: This table connects a patient to a specific game.

values:

- game_id – the id of the game.
- patient_id – the id of the patient.

shapes: This table keeps records of reaction time for each shape in the games.

values:

- game_id – the id of the game.
- arrow – the reaction times for the arrow shape.
- rectangle – the reaction times for the rectangle shape.
- diamond – the reaction times for the diamond shape.
- pie – the reaction times for the pie shape.
- triangle – the reaction times for the triangle shape.
- heart – the reaction times for the heart shape.
- flower – the reaction times for the flower shape.
- hexagon – the reaction times for the hexagon shape.

- moon – the reaction times for the moon shape.
- plus – the reaction times for the plus shape.
- oval – the reaction times for the oval shape.
- two_triangles – the reaction times for the two_triangles shape.
- circle – the reaction times for the circle shape.
- star – the reaction times for the star shape.

textures: This table keeps records of reaction time for each texture in the games.

values:

- game_id – the id of the game.
- four_dots – the reaction times for the four_dots texture.
- waves – the reaction times for the waves texture.
- arrow_head – the reaction times for the arrow_head texture.
- strips – the reaction times for the strips texture.
- happy_smiley – the reaction times for the happy_smiley texture.
- spikes – the reaction times for the spikes texture.
- dollar – the reaction times for the dollar texture.
- net – the reaction times for the net texture.
- note – the reaction times for the note texture.
- arcs – the reaction times for the arcs texture.
- monitor – the reaction times for the monitor texture.
- sad_smiley – the reaction times for the sad_smiley texture.
- strudel – the reaction times for the strudel texture.
- four_bubbles – the reaction times for the four_bubbles texture.
- spiral – the reaction times for the spiral texture.
- squares – the reaction times for the squares texture.

שאלות

שאלות לטבלת patient:

טבלה זו מרכזת את כל המידע האישי על מטופלים.

הטבלה כוללת את הערכים הבאים:

1. תעודת זהות של המטופל
2. שם המטופל
3. תאריך הלידה של המטופל
4. מין המטופל
5. היד הדומיננטית של המטופל
6. סוג המטופל (האם הוא נסיין או לא)

➤ `Select * from patient where patient_id=<patient id>`

השאלתה הזו מחזירה את כל המידע עבור מטופל מסוים בהינתן תעודת זהות שלו.

➤ `Select * from <table name>, patient_game where patient_game.patient_id=<patient id> and patient_game.game_id=<table name>.game_id`

בהינתן תעודת זהות של המטופל ושם של טבלה (shapes/textures) השאלתה מחזירה את כל התוצאות של המשחקים אותם שיחק המטופל (מסוג המשחק שנבחר).

כלומר, עבור הטבלה shapes ות.ז. 1234 השאלתה תחזיר את כל תוצאות המשחקים מסוג shapes ששיחק המטופל שת.ז. שלו היא 1234.

השתמשנו בטבלה המקשרת patient_game כיוון שהיא הטבלה שממפה בין תעודות זהות של המטופלים ומספרי המשחקים שהם שיחקו.

➤ `insert into patient (patient_id, patient_name, patient_gender, dominant_hand, birth_date, patient_type) values (<patient id>, <patient name>, <patient gender>, <dominant hand>, <birth date>, <patient type>)`

השאלתה הזו מכניסה מידע עבור מטופל חדש לטבלה patient. השאלתה מקבלת את המידע שלה מעמוד הרישום (login) של האפליקציה.

➤ Select patient_id from patient

השאלתה מחזירה את כל תעודות הזהות של המטופלים שקיימות במסד הנתונים.
השאלתה נמצאת בשימוש בעיקר בעת יצירת גרפים רוחביים עבור כלל המטופלים.

שאלות לטבלת game:

טבלה זו מרכזת את כל המידע של המשחקים.
הטבלה כוללת את הערכים הבאים:

- מספר המשחק
- סוג המשחק (צורות/ מרקמים/ משולב)
- מספר התמונות שנוחשו נכון
- התאריך שבו התבצע המשחק
- האם המטופל השתמש ביד הדומיננטית
- מגבלת הזמן במהלך המשחק

➤ Select game.game_id, game.game_type, game.num_recognized_button, game.game_date, game.time_limit, game.dominant_hand from game, patient_game, patient where patient.patient_id="<patient id>" and patient.patient_id=patient_game.patient_id and patient_game.game_id=game.game_id

בהינתן ת.ז. של מטופל השאלתה מחזירה את כל המידע על המשחקים ששיחק אותו מטופל.
השתמשנו בטבלה המקשרת patient_game כיוון שהיא הטבלה שמקשרת לנו בין מספרי המשחקים לבין תעודות הזהות של המטופלים.

➤ select * from "<table name>" where game_id=<game id>

בהינתן מספר משחק כלשהו וסוג משחק (shapes/textures) השאלתה הזו מחזירה את תוצאות המשחק מאותו סוג עבור אותו מספר משחק.
כלומר, עבור משחק מספר 12 וטבלה textures השאלתה תחזיר את כל התוצאות של משחק מספר 12 מסוג textures.

➤ insert into game (game_type, num_recognized_buttons, game_date, time_limit, dominant_hand) values ("<game type>", <number of recognized images>, now(), <current game's time limit>, "<did the user used his/her dominant hand>")

השאילתה מכניסה את הגדרות המשחק החדש שהסתיים. השאילתה נקראת בכל סיום משחק ומקבלת את המידע שלה מחלון המשחק.

- insert into patient_game (patient_id, game_id) values ("**<patient id>**", **<game id>**)

השאילתה הזו מכניסה צמד חדש של משחק – מטופל (בתעודות זהות), כדי שנוכל לדעת לקשר בין משחק למטופל ששיחק.

- insert into **<table name(shapes/textures)>** (**<table columns>**) values (**<table values>**)

השאילתה מכניסה את תוצאות המשחק הנוכחי לטבלה המתאימה לפי סוג המשחק. את הערכים השאילתה מקבלת מחלון המשחק שבדיוק הסתיים.

למשל עבור משחק מסוג "משולב" השאילתה תכניס את תוצאות המשחק גם לטבלה shapes וגם לטבלה textures (כל טבלה והערכים המתאימים לה).

Code Structure

ארכיטקטורה: החלטנו להשתמש בארכיטקטורת MVC על ידי חלוקה ל Controller , Model, View.

שפת פיתוח: על מנת לממש חיבור למסד הנתונים החלטנו להשתמש ב-JAVAFX, נודע לנו שהאפליקציה לא תהיה מקוונת ולכן לא פיתחנו אותה בכלי Web. את שאר הקוד של האפליקציה מימשנו ב-JAVA למעט החלק של הייצוא לאקסל אותו מימשנו ב-PYTHON (הרחבה בהמשך).

GUI

תיאור כללי:

ה GUI package בנוי מסט של מחלקות ודפים. האפליקציה תומכת בשתי שפות (עברית ואנגלית) לכן עבור כל חלון יש לנו שני דפי fxml (אחד לכל שפה), מחלקה של controller ודף CSS שנותן לכל ה-fxml תבנית אחידה. דף ה fxml מכיל את שלד החלון כפי שיוצג. מחלקת ה controller שולטת על הפיצ'רים השונים בדף ה fxml ומקשרת אותם לפונקציות מתאימות. דף ה css אחראי על עיצוב החלון. כל החלונות נפרשים על גבי המסך כולו.

פירוט המחלקות:

:Main

זוהי מחלקת ה main של האפליקציה כולה. המחלקה מעלה את דף הפתיחה של האפליקציה, יוצרת את החיבור ל Controller ונותנת חיווי למשתנה במקרה של תקלה. בנוסף, בעת סגירת האפליקציה במחלקה זו יסגר גם החיבור.

:MainWindow

זוהי דף הפתיחה של האפליקציה. האפליקציה נפתחת בשפה האנגלית כאשר בפינה השמאלית העליונה יש דגל ישראל שמאפשר מעבר לשפה העברית (באופן מקביל בדף בעברית יש דגל ארה"ב שמאפשר בלחיצה לעבור לשפה האנגלית). למשתמש יש 4 אפשרויות בדף זה:

1. משתמש חדש- יוכל להירשם (מעבר ל - SignUpWindow).
2. משתמש קיים- יוכל להכניס ת.ז במקום המתאים ולהתחבר (במידה והת.ז אכן במערכת, אחרת Alert מעבר ל- MenuWindow).
3. אדמין- יוכל להכניס שם משתמש וסיסמא ולעבור לאזור של האדמין (במידה והפרטים אכן נכונים, אחרת Alert מעבר ל- AdminWindow).
4. יציאה מהאפליקציה- בלחיצה על כפתור היציאה.

:SignUpWindow

מבקש את הפרטים הבאים מהמשתמש: שם פרטי, שם משפחה, 4 ספרות אחרונות של ת.ז, מין, תאריך לידה, יד דומיננטית והאם אתה מטופל או נסיין (קבוצת ביקורת). לאחר וידוא כי הוכנסו כל הפרטים ובפורמט הנכון נוצר PatientContainer מתאים ומבצעים שאילתת הכנסה למסד הנתונים. כמו כן, ישנו כפתור חזרה המאפשר לחזור לחלון הראשי (MainWindow).

:MenuWindow

תפריט עבור משתמש רשום. ישנו זיהוי לפי השם בחלק העליון של הדף. מאפשר מעבר לדף הוראות (InstructionsWindow), מעבר להתחלת משחק (SettingsWindow), מעבר לאזור האישי (GraphsWindow) ויציאה מהאפליקציה.

כמו כן, ניתן ללחוץ על התמונה של הדלת למעלה וכך למעשה להתנתק ולחזור לחלון הפתיחה (MainWindow).

:InstructionsWindow

הוראות עבור המשחק, נקרא מהקובץ המתאים לשפה שנמצא בresources. ישנו חץ חזרה שמאפשר להגיע בחזרה לתפריט (MenuWindow) ותמונה של דלת שמאפשרת להתנתק ולחזור לחלון הפתיחה (MainWindow). ישנו זיהוי לפי השם בחלק העליון של הדף.

:SettingsWindow

בחלון זה נגדיר את הגדרות המשחק (סוג משחק, הגבלת זמן, משחק ביד דומיננטית או לא) ונעבור למשחק עצמו (GameWindow). כמו כן, ישנו חץ חזרה שמאפשר להגיע בחזרה לתפריט (MenuWindow) ותמונה של דלת שמאפשרת להתנתק ולחזור לחלון הפתיחה (MainWindow). ישנו זיהוי לפי השם בחלק העליון של הדף.

:GameWindow

מציג את המשחק המתאים להגדרות שנבחרו, נותן חיווי של תמונה וצליל עבור זיהוי נכון\ לא נכון. כמו כן, נותן חיווי עבור הצלחה רצופה של 1\3, 2\3 מהתמונות ברצף (תמונה של מחיאות כפיים וצליל מתאים). מציג את הזמן שנותר לזיהוי התמונה וכמות התמונות שעברו מבין כלל התמונות שיש לזהות. בסוף עובר לדף התוצאות (ResultsWindow). ישנו זיהוי לפי השם בחלק העליון של הדף. כמו כן, ישנו חץ חזרה שמאפשר להגיע בחזרה לתפריט (MenuWindow) ותמונה של דלת שמאפשרת להתנתק ולחזור לחלון הפתיחה (MainWindow).

:ResultsWindow

מציג ממוצע זמן תגובה של המשחק ומספר צורות שנחששו נכונה. מאפשר מעבר לאזור האישי (GraphsWindow) ויציאה מהאפליקציה. ישנו זיהוי לפי השם בחלק העליון של הדף. כמו כן, ישנו חץ חזרה שמאפשר להגיע בחזרה לתפריט (MenuWindow) ותמונה של דלת שמאפשרת להתנתק ולחזור לחלון הפתיחה (MainWindow).

:GraphsWindows

מכיל 3 סליידים, הראשון מציג את פרטי המשתמש, השני את טבלת התוצאות והשלישי גרפים. אציג כל אחד מהם בנפרד. כמו כן, מאפשר לחזור אחורה בלחיצה על החץ (לתפריט או לאזור אדמין תלוי מאיפה הגענו) ולהתנתק ולחזור לדף הראשי בלחיצה על הדלת. כמו כן מכיל זיהוי של המשתמש בחלק העליון של המסך.

- פרטי משתמש: מציג את הפרטים איתם המשתמש נרשם ומאפשר לייצא נתונים לטבלת אקסל.
- טבלת תוצאות: מציג את התוצאות של כל המשחקים, זמן לזיהוי כל צורה, סוג המשחק וכו'. כמו כן, מאפשר בלחיצה כפולה על שורה לראות גרף מתאים.
- גרפים: מאפשר לעקוב אחר ההתקדמות בזיהוי במשחק צורות ובמשחק מרקמים לאורך זמן.

:AdminWindow

מכיל מעבר חזרה (בעזרת החץ) לדף הראשי (MainWindow).

מציג טבלה המכילה פרטים של מטופלים, לחיצה כפולה על שורה של מטופל תעביר לאזור האישי שלו (GraphsWindow). כמו כן ניתן ללחוץ על כפתורי הגרפים ולקבל גרף רוחבי מתאים לחתך האוכלוסיה ולסוג המשחק הנבחר.

:BasicWindow

מחלקת בסיס ממנה יורשות חלק מהמחלקות האחרות. מכילה את הקוד של יציאה מהאפליקציה, קוד של מעבר לתפריט הראשי (MenuWindow) והקוד שמתבצע כאשר מתנתקים ומועברים לחלון הפתיחה (MainWindow).

:HoveredThresholdNode

מחלקה שמשמשת למטרה עיצובית. מאפשרת בעת ריחוף על הגרפים בתפריט האישי (GraphsWindow) לראות חלון קטן עם התוצאה המדויקת בנק' שריחפנו מעליה בגרף.

Model

תיאור כללי:

המודל הוא החלק אשר מתקשר עם ה- DataBase . הוא היחיד שיוצר איתו קשר ומנהל את החיבור אליו. המודל מקבל מידע מה Controller עם פרמטרים שאיתם הוא אמור לבצע שאילתות ל- DataBase.

פירוט המחלקות:

:Connection

מחלקה זו היא למעשה Singleton כי אנו מאפשרים יצירה של מופע אחד שלה בלבד. כל בקשה (מעבר לבקשה הראשונית) תקבל את המופע הקיים. המחלקה יוצרת את ה controller ים השונים שבשימוש (patientController, gameController). כיוון שהמחלקה היא סינגלטון גם מחלקות אלה יוצרו רק פעם אחת. כאשר פונים אל מחלקה זו עם בקשה, היא מתווכת את הבקשה לcontroller המתאים ומחזירה את התוצאות.

:Model

האינטרפייס של המודל. מבטא את הפונקציונליות של המודל. כל מי שמרחיב את האינטרפייס צריך לממש מספר פונקציות , ביניהן:

- openConnection
- closeConnection
- getData

:DBModel

המחלקה העיקרית של המודל. מחלקה זו מקבלת מהקונטרולרים את המידע הנצרך לשאילתות. המודל בודק מידע זה ועל פיו קורא לפונקציות המתאימות במחלקות שמחזיקות את הפונקציות שמבצעות את השאילתות. המודל:

- מנתב את המידע שמתקבל לפונקציות המתאימות על מנת לבצע אותן.
- מחזיר את המידע שהתקבל מביצוע השאילתות ושולח בחזרה לקונטרולרים.

:ExcelWriter

קובץ פייתון שמטרתו ייצא טבלאות וגרפים לקובץ אקסל. הקובץ מקבל את המידע של הטבלה ושל שני הגרפים מהאזור האישי, המידע מופרד ומוזרם לקובץ אקסל שם הם נבנים מחדש. הקובץ מורץ ע"י התוכנית תוך הפעלת script שרץ ב-process נפרד.

:Executor

המחלקה מוציאה לפועל את השאילתות. כל מחלקה שמחזיקה פונקציות של שאילתות בונה את השאילתות ושולחת אותם למחלקה הזו. המחלקה מבצעת את השאילתה ומאגדת את הנתונים שחוזרים בצורה נוחה לשימוש.

:GameQueries

המחלקה אחראית על ביצוע כל השאילתות הקשורות למשחקים. המודל מתקשר עם המחלקה הזו וקורא לפונקציות הנצרכות שלה בהתאם לפרמטרים נתונים. המחלקה מחזיקה פונקציות שכל אחת מהן היא שאילתה נצרכת לטבלה.

:PatientQueries

המחלקה אחראית על ביצוע כל השאילתות הקשורות למטופלים.
המודל מתקשר עם המחלקה הזו וקורא לפונקציות הנצרכות שלה בהתאם לפרמטרים נתונים.
המחלקה מחזיקה פונקציות שכל אחת מהן היא שאילתה נצרכת לטבלה.

Resources – מחלקות

תיאור כללי:

זו החבילה שבה נמצאות כל המחלקות המשותפות בין החלקים השונים, כגון מחזיקי מידע המשמשים להעברת נתונים.

פירוט המחלקות:

:Alerter

מטרת המחלקה היא לספק התראות שונות עבור מצבים שונים.
במידת הצורך החלונות השונים ישתמשו בהתראות המתאימות האלו. למשל, התראה על חוסר יכולת להתחבר לDB.

:AdminTableInfoContainer

המחלקה מאגדת את כל המידע הנדרש בשביל להציג בטבלה הנמצאת בעמוד האדמין. הטבלה מכילה ערכים דומים למידע בטבלת ה patient.

:AlertMessages

מטרת המחלקה היא לאגד סוגים שונים של הודעות והתראות שיכולות לצוץ במהלך ריצת האפליקציה.
בכל פעם שנרצה להתריע נבחר אחת מן ההודעות הקיימות במחלקה הזו ונציג אותה למשתמש.

:imageTypeEnum

מטרת המחלקה היא לתת ייצוג לכל אחד מסוגי המשחק האפשריים.
המחלקה מכילה את הסוגים Shapes/Textures/Both.

:GameContainer

מטרת המחלקה היא ליצור אובייקט שמאגד את כל המידע הנוצר ממשחק בודד.
המחלקה מכילה את הערכים הבאים:

- 2 מפות המכילות את זמני התגובה של השחקן עבור כל צורה/ טקסטורה.
כל מפה ממפה בין כותרת התמונה (למשל "arrow" מטיפוס string) לזמן התגובה שלקח עד שהשחקן זיהה את התמונה והקיש על המקש הנכון (מטיפוס double).
- סוג המשחק
- תאריך המשחק
- האם השחקן השתמש ביד הדומיננטית
- מספר התמונות שהשחקן זיהה נכון
- מגבלת הזמן של המשחק

:PatientContainer

מטרת המחלקה היא ליצור אובייקט שמאגד את כל המידע של משתמש מסוים.
המחלקה נוצרת בצורה של singleton, כלומר נוצרת פעם אחת ובכל קריאה מחזירה את אותו אובייקט.
בעת התנתקות והתחברות של משתמש חדש המחלקה מעדכנת את ערכיה עבור המשתמש החדש.

המחלקה מכילה את הערכים הבאים:

- שם המשתמש
- תעודת הזהות של המשתמש
- תאריך הלידה של המשתמש
- מין המשתמש
- היד הדומיננטית של המשתמש
- סוג המשתמש (נסיין/ מטופל)

:PieChartBuilder

מטרת המחלקה היא ליצור גרפי עוגה (Pie Charts) שונים.

המחלקה יוצרת גרפים שונים שמראים אחוזי הצלחה וכשלון של חתכים שונים מתוך המשתמשים על סמך הנתונים שהיא מקבלת.
החתכים הקיימים הם:

- צורות – מטופלים
- טקסטורות – מטופלים
- משולב – מטופלים
- צורות – נסיינים
- טקסטורות – נסיינים
- משולב – נסיינים

:DBConnectionException

שגיאה חדשה שמטרתה להתריע על חיבור כושל למסד הנתונים.

:TableInfoContainer

המחלקה מכילה את כל הפרטים וההגדרות עבור משחק ספציפי בנוסף לתוצאות המשחק. כל אובייקט כזה מייצג בסופו של דבר שורה בטבלה השוכנת באזור האיש.

Resources – קבצים

תיקיית pic:

מכילה 3 תתי תיקיות של תמונות:

1. Misc – כל התמונות שקשורות לעיצוב של האפליקציה.
2. Shapes – כל התמונות של הצורות שיוצגו במשחק למשתמש.
3. Textures – כל התמונות של המרקמים שיוצגו במשחק למשתמש.

תיקיית sounds:

מכילה 3 צלילים שנמצאים בשימוש במשחק:

1. Correct_Answer_Sound_Effect
2. Wrong_Answer_Sound_Effect
3. Applause

admin_details:

שם המשתמש והסיסמא איתם האדמין מתחבר לאזור שלו.

config:

קובץ הקונפיגורציה מכיל את הפרטים הדרושים על מנת להתחבר למסד הנתונים: user, port, host, password, scheme.

instructionsFileEng:

קובץ ההוראות בשפה האנגלית.

instructionsFileHeb:

קובץ ההוראות בשפה העברית.

shapesToKeys:

מיפוי בין השמות של התמונות מסוג צורה למקש המתאים במקלדת.

texturesToKeys:

מיפוי בין השמות של התמונות מסוג מרקם למקש המתאים במקלדת.

Controller

תיאור כללי:

מטרת המחלקות בחבילה הזו היא לגשר בין המחלקות שמפעילות את התצוגה הגרפית לבין המחלקות שנמצאות ב Model ואחריות על לוגיקה ופעולות מול מסד הנתונים. ה Controller מעביר בקשות ומידע מה GUI אל ה Model ולהיפך.

פירוט המחלקות:

:ControllerInterface

כל מחלקות שמממשות את הממשק הזה חייבות לממש 2 מתודות:

```
void openModelConnection() throws SQLException;  
void closeModelConnection() throws SQLException;
```

2 המתודות הללו אחראיות על פתיחת וסגירת החיבור עם מסד הנתונים.

:ControllerAbstract

זו מחלקה אבסטרקטית המממשת את הממשק ControllerInterface. תפקידה של המחלקה הוא החזקת אובייקט מסוג Model וכן פתיחת וסגירת החיבור שלו. בנוסף, כל מחלקה היורשת ממחלקה זו תהיה רשאית לגשת לאובייקט ה Model שלה ולהשתמש בו לצרכיה. המחלקות PatientController ו GameController יורשות מהמחלקה האבסטרקטית. כל אחת מהן אחראית לקשר בין ה GUI לבין השאלות המתאימות לה.

בכל אחת מהמחלקות המרחיבות את המחלקה האבסטרקטית, כך מתבצע אופן החיבור:

- מה-GUI ל-GUI Controller – ה GUI מעביר ל controller את המידע שהוא צריך ולפיו הוא יודע איזו שאלתה צריך.
- מה-GUI Controller ל-Model – ה Controller משתמש במודל בשביל ליצור אובייקט Query לפי סוג השאלתה המתאימה (PatientQueries/ GameQueries) ומכניס את המידע שקיבל מה GUI אל ה-Query בשביל לבצע שאלתה על מסד הנתונים.
- מה-Model ל-GUI Controller – המידע שחוזר מה Model נשמר בסוגים שונים של מבני נתונים מתאימים ומועבר ל Controller.
- מה-GUI Controller ל-GUI – המידע עובר עיבוד סופי ומועבר מה Controller אל ה GUI.

ספריות חיצוניות

JFoenix: ספריה שמכילה אובייקטים מעוצבים של JavaFX. החלטנו להשתמש בספריה זו בשביל לעצב את האפליקציה באופן יותר חדשני ומעניין.

השתמשנו ב 2 חבילות חיצוניות בשביל החיבור של java ל-SQL:

1. JDBC MySQL

2. MySQLConnector

Openpyxl: ספריית פייתון המאפשרת אינטגרציה בין פייתון Office Excel כמו גם יצירת קבצי Excel עם מידע מהתוכנית.

הספריה עזרה לנו בייצוא המידע מהאפליקציה לקבצי Excel בנוסף עזרה ליצירת גרפים במסמך.