**README**

מגישים:   
אופק כהן, 207642224, ofekco@mta.ac.il

ניב גורסקי, 206094914, [nivgr@mta.ac.il](mailto:nivgr@mta.ac.il)

**בונוסים:**

מימושנו את הבונוס של החלפת ה-skin של המערכת. מימשנו 3 ערכות צבעים נוספות לערכה הדיפולטית הטעונה כבר במערכת. ניתן להחליף את ערכת הצבעים באמצעות combo box שנמצא בראש החלון (display mode).

הנחות:

1. יצאנו מנקודת הנחה שיתרת חשבון יכולה להיות מספר לא שלם (כמו בחיים האמיתיים) ולכן התייחסנו אליו גם ככזה (double) וכתוצאה מכך גם לסכומי תשלום להלוואה ולריביות.

תיעוד המערכת:

**מודול המנוע (Engine):**

1. מחלקת Customer- מחלקה המכילה את המידע הבסיסי על לקוח (שמו והחשבון שלו), וכן את אוסף ההלוואות אליהן הלקוח משויך כלווה ואת אוסף ההלוואות אליהן הלקוח משויך כמלווה.
2. מחלקת Loan- מחלקה המכילה את כל המידע הרלוונטי להלוואה. בנוסף, בתוך כל הלוואה ישנו מידע עבור התשלומים של ההלוואה (מופע של LoanPaymentsData)
3. Account – מחלקה המכילה מידע על חשבון של לקוח במערכת (יתרה ותנועות בחשבון).
4. LoanPaymentsData – מחלקה המכילה מידע אודות התשלומים של הלוואה. המחלקה מאגדת בתוכה את המחלקה Payment שמייצגת תשלום בודד, וכן את מבני הנתונים עבור כל סוגי התשלומים המקושרים להלוואה (תשלומים אשר שולמו, טרם שולמו, ושעוכבו).
5. מחלקת Timeline- מחלקה המכילה את הי"ז הנוכחי של המערכת ומאפשרת לקדם אותו או לאפס את הי"ז של המערכת בעת טעינת קובץ חדש.
6. במודול זה יצרנו מספר מחלקות שתפקידן לקרוא נתונים מקובץ ה-XML לפי הסכימה הנתונה, לחלץ ממנו אובייקטים ולהמירם לאובייקטים המוגדים במערכת שלנו. מחלקות אלה נמצאות תחת ה-package XML\_Handler"".

במודול זה ייצרנו 3 ממשקים:

* paymentsDB: ממשק זה מתייחס לאופן אחסון נתוני התשלומים במערכת.   
  הרעיון הוא לאפשר חלוקה נוחה של סוגי התשלומים השונים (תשלומים אשר שולמו, טרם שולמו, ושעוכבו), תוך מידור מחלקת ההלוואה לאופן בו נשמרים התשלומים בפועל ולחלוקה ביניהם. הדבר מאפשר להוסיף סוגי תשלומים חדשים או לשנות את מבני הנתונים בהם נשמרים התשלומים מבלי לשנות דבר בהלוואה עצמה.  
  המחלקות: ExpiredPaymentsByYaz, PayedPaymentsByYaz, UnpaidPaymentsByYaz ממשות את הממשק הנ"ל.
* MainSystem: ממשק זה מהווה את אמצעי הגישה של ה-UI למערכת. ה-UI מכיר את המערכת כמערכת כללית כלשהי וחשוף רק לפעולות אותן הוא מאפשר למשתמש לבחור לבצע במהלך האינטראקציה שלו עם המשתמש, ובכך מתקבלת נוחות ובטיחות שימוש במערכת שכן ל-UI לא אכפת באיזה אופן מבצעת המערכת את הפעולות המוגדרות למשתמש. בעתיד אם נרצה לתמוך בפעולות נוספות עבור המשתמש נוכל להוסיפה לממשק זה ולממש אותה במערכת ביתר קלות.
* systemService: ממשק זה מהווה מעטפת החושפת מתודות שירות של מערכת ההלוואות כולה למחלקות האבסטרקטיות האחראיות על שיבוץ ההלוואות ללקוחות, והזזת ציר הזמן וביצוע התשלומים. הרעיון הוא לא לחשוף למחלקות אלו את כל המידע הפנימי והמתודות של מערכת ההלוואות כולה אלא רק את פעולות השירות הנחוצות להן מהמערכת.

בעתיד אם נצטרך לממש פעולות נוספות בדומה לפעולות השיבוץ וביצוע התשלומים נוכל ביתר קלות להוסיפן לממשק ולממש אותן במערכת.

**מודול ה-JavafxUI:**

מחלקות ה-UI המכילות את קבצי ה-fxml , הקונטרולרים של רכיבי ה-UI וקבצי ה-CSS.

**מודול ה- DTO:**

במודול זה ישנן מחלקות שונות שתפקידן להוות container של מידע שמועבר מהמנוע ל-UI.   
מחלקות אלו מייצגות אובייקטים מרכזיים במנוע (ולעיתים אף שילוב שלהם לפי הדרישה של כל בקשת משתמש), והן אלו שמאפשרות העברת מידע מהמנוע ל-UI מבלי שה-UI יהיה חשוף לאובייקטים הפנימיים וללוגיקה הפנימית המורכבת של המערכת.

**מודול ה- EXCEPTIONS:**

במודול זה ישנן מחלקות exceptions ייעודיות למערכת.