**מערכת בנקאות אלטרנטיבית תרגיל 3**

שמות המגישים:

1. איתמר פודולסקי , ת"ז 206026254
2. איתי לוי , ת"ז 206556698

מיילים זמינים לשאלות : [itailv@mta.ac.il](mailto:itailv@mta.ac.il), [itamarpo@mta.ac.il](mailto:itamarpo@mta.ac.il)

**הערה:** יש לנו אישור דחייה של ההגשה ב3 ימים בעקבות מילואים, אישור 3010 יצורף.

**בונוסים שמומשו:**

1. 🡨POSTMAN לינק לcollection נמצא בתיקיית ההגשה.

לשימוש בPOSTMAN יש לעשות IMPORT לקובץ הPOSTMAN המצורף.

הסבר מפורט על הפעלת הבקשות הרלוונטיות לבונוס ובכלל בעמודים **3-5.**

1. צ'אט. (לא נתמך על ידי POSTMAN)

**הנחות שביצענו על דעת עצמנו:**

1. א. אופן חלוקת הכספים של הלקוח בעת שיבוץ: אם סכום כל ההלוואות שבחר הלקוח אינו עולה על הסכום שהוא בחר להשקיע, הוא פשוט ישקיע בכל הלוואה את הסכום שחסר לה כדי להפוך לactive

אחרת, הוא יחלק את הכספים בצורה שווה ככל הניתן, לפי אלגוריתם שמימשנו במנוע.

ב. מבחינתנו סכום השקעה אינו פרמטר רלוונטי לפילטור, ולכן במידה ויש בעיה עם סכום ההשקעה המשתמש יקבל על כך מידע רק כאשר ינסה לבצע את השיבוץ. הדבר נכון גם לאחוז בעלות על הלוואה.

1. במספרים עשרוניים החלטנו להדפיס עד 2 מספרים אחרי הנקודה העשרונית.
2. המשתמש יכול להכניס בסינון של הריבית בשיבוץ איזה מספר חיובי שהוא רוצה, שכן בעולם האמיתי יש דבר כזה ריבית של יותר ממאה אחוז.
3. כאשר ההלוואה נמצאת בrisk , המשתמש **חייב** לבחור כמה מהתשלום שהוא חייב הוא מעוניין לכסות. מבחינתנו, אם הוא הכניס יותר מגובה התשלום, אנחנו סוגרים את התשלום עבורו וזו אינה שגיאה.
4. כאשר המערכת במצב REWIND, הלקוח מקבל גישה אך ורק לטאב המידע, וכל שאר הטאבים חסומים לו לחלוטין.

**פירוט המחלקות והמודולים:**

1. JAVAFX- מודול זה אחראי על ניהול כל הרכיבים והמסכים של האפליקציה, ופנייה למודולי שתי האפליקציות (אדמין ולקוח) כדי שיצרו בקשות http להשגת מידעים וביצוע פעולות.

הרכיבים מתחלקים ל4 ספריות עיקריות, שהן:

1. Admin – מכיל את כל הרכיבים שקשורים לאפליקציית האדמין, למעט מסך הlogin.
2. Customer – מכיל את כל הרכיבים הקשורים לאפליקציית הלקוח, ביניהם מסך המידע, השיבוץ, התשלום ,מכירת\קניית הלוואה ויצירת הלוואה חדשה.
3. Table – מכיל את כל סוגי הטבלאות בהם אנו משתמשים במהלך התכנית.
4. ENGINE – מודול זה אחראי על שמירת כל נתוני המערכת, וביצוע הלוגיקה מאחורי הפעולות. ישנו מופע של ENGINE בתוך הUI. מחלקות עיקריות בENGINE:
5. Loans - מחלקה המחזיקה את כל הנתונים הקשורים להלוואות של לקוחות.

מחלקות קשורות: Payment – מחלקה המחזיקה נתונים בנוגע לתשלומי הלוואות

LoanStatus – מחלקה המחזיקה נתונים בנוגע לסטטוס הלוואה קיימת.

1. Customer – מחלקה המחזיקה מידע על לקוחות קיימים, כוללת שם לקוח, יתרה, הלוואות שהלקוח משויך אליהן, והיסטוריית פעולות בחשבון.

מחלקות קשורות: AccountTransaction - מחלקה המחזיקה פירוט על פעולה בחשבון של לקוח מסוים.

1. DTO – מודול זה משמש להעברת מידע בין הENGINE ורכיבי הUI, תוך מעבר בסרבלטים השונים של השרת. הוא אינו מבצע שום פעולות לוגיות, ותפקידו הוא אך ורק להעביר מידע.

פקגי'ם רלוונטיים:

1. Exceptions – מכיל מחלקות שזורקות exception מתאים , במקרים כמו קלט לא תקין, משיכת סכום לא תקין של כסף מחשבון של לקוח , אי תקינות של קובץ xml וכו'.
2. loansDTO – מכיל מחלקות המכילות את המידעים הרלוונטיים לאופציה 2 בתפריט, לפי סטטוס ההלוואה.
3. customerDTO – מכיל מחלקות המכילות את המידעים הרלוונטיים לאופציה 3 בתפריט, לפי הלקוחות השונים הקיימים במערכת.
4. מחלקות שייעודן הוא להעביר מידעים תוך שימוש בJSON על גבי בקשות HTTP.
5. AdminApp - המודול האחראי על ניהול אפליקציית האדמין. מכיל שתי מחלקות עיקריות, הראשונה האחראית על מסך הlogin של האדמין, ומכיל גם את רכיבי הUI שלו.

השנייה אחראית על ניהול אפליקציית האדמין לאחר הכניסה למערכת, מתקשר עם רכיבי הUI של מסך האדמין הנמצאים במודול JAVAFX, ואחראי על ייצור בקשות HTTP ותקשורת עם השרת.

1. CustomerApp – המודול האחראי על ניהול אפליקציית הלקוח. גם הוא פועל באותה צורה, שתי מחלקות עיקריות, אחת למסך הlogin ואחת שאחראית לדבר עם רכיבי הUI והשרת.
2. Server – המודול המכיל את הסרבלטים של השרת. מחולק ל3 פקג'ים שונים:
3. Admin : מכיל את כל בקשות הHTTP שהאדמין מבקש לבצע, ביניהן הכנסת והוצאת המערכת ממצב REWIND, הזזת ציר הזמן לאחור, קידום י"ז המערכת וקבלת כל מידעי המערכת.
4. Common : מכיל את הבקשות המשותפות גם לאדמין וגם ללקוח, במקרה של המערכת שלנו, מכיל רק את בקשת הlogin.
5. Customer : מכיל את כל בקשות הHTTP שלקוח מבקש לבצע. מחולק לפקג'ים לפי הטאבים השונים, ובכל אחד מהם מחלקות המיועדות לביצוע כל הפעולות הנדרשות לתפעול הטאב.

**הסבר כללי על אופן השימוש במערכת:**

**מהרגע שבו לקוח כלשהו התחבר למערכת, נוצר מופע של המנוע וספירת הי"ז מתחילה מ1.**

**כל עוד המערכת אינה במצב REWIND (מצב ACTIVE) כל המשתמשים יכולים לבצע את כל הפעולות העומדות לרשותן ללא הגבלה. עם זאת, במצב REWIND ייחסמו כל האופציות שעשויות לגרום לשינוי מידעי המערכת, גם אצל האדמין וגם אצל הלקוח, והמערכת תעבור למצב READ ONLY.**

**אפליקציית האדמין**

בעת טעינת האפליקציה, המשתמש מתבקש לבחור שם (יכול לבחור כל שם שבא לו שעוד לא נמצא במערכת) ולאחר מכן מועבר למסך האדמין, שם הוא יכול לראות את מידעי המערכת, לקדם את הזמן ולהזיז את ציר הזמן לאחור. לפי המימוש שלנו, לא חייב להיות אדמין במערכת, לקוחות יכולים להתחבר כמה שבא להם, אבל כל עוד אין אדמין הם יהיו תקועים בי"ז 1. כמו כן, רק אדמין אחד יכול להתחבר בכל רגע נתון (דרך אפליקציית האדמין).

**אפליקציית הלקוח**

בעת טעינת האפליקציה, המשתמש מתבקש לבחור שם (יכול לבחור כל שם שבא לו שעוד לא נמצא במערכת) ולאחר מכן מועבר למסך הלקוח, שם מוצגים בפניו חמשת הטאבים כפי שמוגדרים כאן למטה ובנוסף סרגל כלים בחלק העליון של המסך המכיל את שם הלקוח, י"ז נוכחי, כפתור לטעינת קובץ ושינוי SKIN למערכת.

1. Information – במסך זה ישנם מידעים על הלוואות שבהן הלקוח לוקח חלק. הטבלה העליונה מתארת הלוואות בבעלותו, והתחתונה הלוואות שלהן הוא מלווה.

בצד ימין של המסך ישנה טבלת המציגה את פעולותיו של הלקוח בחשבון. בתחתית מסך זה ישנם שני כפתורים של משיכת\הפקדת כספים, שלחיצה עליהם תוביל לפתיחת חלון פופ-אפ שבו המשתמש צריך להכניס סכום כסף רצוי.

1. Inlay – זהו מסך השיבוץ. בחלקו העליון, המשתמש צריך להכניס את הפרמטרים לפיו הוא היה רוצה שהשיבוץ יתבצע. למעט סכום ההשקעה שהוא מחייב, כל שאר הפרמטרים אינם חובה, ובשביל לבחור בהם צריך לסמן בוי את הcheckbox הרלוונטי ולהכניס ערך. באופציית הקטגוריות, בחירה ריקה משמעותה בחירה של כל הקטגוריות.

לאחר הכנסת הפרמטרים יש ללחוץ על כפתור הconfirm בתחתית החלק העליון של המסך.

לאחר הפילטור, יוצגו למשתמש בתחתית המסך בטבלאות ההלוואות שבהן הוא יכול להשקיע, בדמות טבלה של הלוואות מסוג new וטבלה של הלוואות מסוג pending. עליו לבחור את ההלוואות הרצויות ע"י לחיצה על הcheckbox המתאים, ולאחר מכן לחיצה על כפתור הconfirm selection בתחתית המסך.

1. Payments – במסך זה ישנן 2 אפשרויות : ביצוע תשלום בודד או סגירת הלוואה.

ביצוע תשלום בודד – במסך זה שתי טבלאות 🡨 הלוואות active והלוואת risk.

בהלוואות הactive יופיעו ההלוואות שמועד התשלום שלהן הגיע היום, ואילו בrisk יוצגו בעצם כולן, כי כאשר הלוואה נמצאת בסיכון המשתמש יכול לשלם אותה בכל י"ז שיבחר, ולא רק כאשר הגיע מועד התשלום שלה. עבור תשלום הלוואה בסיכון המשתמש חייב להכניס סכום שאותו הוא רוצה לשלם, ונפתחת עבורו האופציה מעל כפתור התשלום שבתחתית המסך.

סגירת הלוואה במכה- במסך זה יופיעו כל ההלוואות שבבעלות הלקוח, בשתי טבלאות לפי סטטוס. המשתמש יוכל לבחור הלוואה, וללחוץ על הכפתור בתחתית המסך כדי לסגור אותה.

בכל חלקי המסך הזה, המשתמש יכול לבחור **רק הלוואה אחת** בכל פעם לביצוע פעולה כלשהי, ולא שתיים או יותר במקביל, כאשר הבחירה נעשית ע"י סימון השורה הרלוונטית בטבלה.

1. Buy\Sell loans – במסך זה ניתנת האפשרות ללקוחות למכור את חלקם בהלוואה מסוימת או לקנות חלק של מישהו אחר בהלוואה שאינם בעלים בה.

במסך המכירה, יוצגו בפני הלקוח הלוואות שהוא יכול להעמיד למכירה, ועליו לבחור את הcheckbox המתאימים וללחוץ על כפתור האישור.

במסך הקנייה, המשתמש יכול לבחור הלוואה אחת בלבד בכל פעם שירצה לקנות. עליו לבחור את השורה הרלוונטית ע"י לחיצה עליה, וללחוץ על כפתור האישור בתחתית המסך.

1. Create new loan - במסך זה ניתנת ללקוח האפשרות להגיש הלוואה חדשה למערכת. הוא יצטרך לבחור שם להלוואה, לבחור קטגוריה, סכום הלוואה , ריבית ומרווח תשלומים. יש לשים לב שזמן הלוואה מתחלק במרווח התשלומים ללא שארית (כמובן שמתבצעת בדיקה לכך).

**הסבר על העבודה עם POSTMAN**

**הcollection שלנו מתחלק ל3 ספריות: admin, common וcustomer.**

**נפרט כאן על הפעולות הנדרשות למימוש הבונוס:**

* **Login** 🡨 מיקום : Servlets/Common. מצפה לקבל שני פרמטרים: שם משתמש, והאם המשתמש הוא אדמין הוא לקוח. לשדה של סוג הלקוח קוראים isAdmin, הכנסת הערך true משמעותה להתחבר כאדמין, והכנסת הערך false משמעותה להתחבר כלקוח רגיל.
* 🡨**UploadFile** מיקום: Servlets/Customer. מצפה לקבל בגוף הבקשה קובץ בצורת form data. יש להעלות את הקובץ הרצוי ליד הפרמטר "file".
* 🡨**CreateNewLoan** מיקום : Servlets/Customer/CreateLoanTab. מצפה לקבל כפרמטרים את כל המידעים הדרושים ליצירת הלוואה. בנוסף בספרייה הנ"ל נמצאת עוד בקשה הנקראת getCategories, המחזירה כקובץ JSON את רשימת הקטגוריות שניתן ליצור עבורן הלוואה.

getCategories לא מצפה לקבל שום פרמטרים.

* 🡨**FilterLoansForInlay** מיקום: Servlets/Customer/InlayTab. מצפה לקבל כפרמטרים את כל המידעים שיכולים להוות סינון בבחירת הלוואה לשיבוץ, כלומר מינימום ריבית, מינימום זמן כולל להלוואה ומקסימום הלוואות פתוחות לבעל ההלוואה. בנוסף, הבקשה מצפה לקבל בגוף הבקשה בצורת raw, כקובץ JSON, את רשימת הקטגוריות לסינון. JSON כזה לדוגמא:

[

    "Setup a business",

    "Happy Event"

]

גם כאן, באותה הספרייה, שמנו את הבקשה getCategories, שכפי שרשום למעלה מחזירה קובץ JSON של קטגוריות המערכת, כך שכל מה שצריך לעשות הוא להעתיק את התשובה שgetCategories מחזירה לגוף הבקשה של סינון ההלוואות, ולמחוק את אלו שלא רוצים.( הערה: אנו מבצעים תמיכה גם בשליחת קובץ JSON ריק, כלומר קובץ הנראה כך : [ ], ומשמעותו בחירת כל הקטגוריות במערכת). התשובה של הבקשה היא מידעי כל ההלוואות העומדות בדרישות הסינון, בצורה של JSON.

* 🡨 **MakeInlay**מיקום: Servlets/Customer/InlayTab. מצפה לקבל כפרמטרים את סכום ההשקעה, ואת אחוז הבעלות המרבית בכל אחת מן ההלוואות שהכסף יושקע בהן. כמו כן, מצפה לקבל בגוף הבקשה בצורת raw, JSON שיכיל את שמות ההלוואות שרוצים להשקיע בהן. JSON כזה לדוגמא:

[

    "Damn the banks",

"stock market introduction"

]

את שמות ההלוואות ניתן לקחת כמובן מהתשובה שחוזרת מבקשת הסינון.

התשובה של בקשה זו היא הודעה המסמנת למשתמש אם השיבוץ עבר בהצלחה או שמה יש תקלות.

**פירוט נוסף על בקשות בpostman שלא נדרשות לבונוס:**

* **עבור אדמין:**

1) 🡨 **IncreaseYaz**מיקום: Servlets/Admin. לא מצפה לקבל פרמטרים, מקדמת את י"ז המערכת. בתשובה חוזר הי"ז החדש.

2) 🡨 **ActivateRewind**מיקום: Servlets/Admin. לא מצפה לקבל פרמטרים. מעבירה את המערכת למצב REWIND. בתשובה חוזר מידע על הצלחת הבקשה.

3) 🡨 **DeactivateRewind**מיקום: Servlets/Admin. לא מצפה לקבל פרמטרים. מורידה את המערכת ממצב REWIND. בתשובה חוזר מידע על הצלחת הבקשה.

4) 🡨 **RewindTime**מיקום: Servlets/Admin. מצפה לקבל כפרמטר את הי"ז אליו אנחנו רוצים לחזור. בתשובה חוזר מידע על הצלחת הבקשה.

5) 🡨 **GetServerInformation**מיקום: Servlets/Admin. לא מצפה לקבל פרמטרים. הבקשה מחזירה בקובץ JSON את כל מידעי המערכת הנגישים לאדמין, כולל מידע על כל הלוואות המערכת, וכל הלקוחות הנמצאים במערכת.

* **עבור לקוח:**

1) 🡨 **GetLoansForSaleAndBuy**מיקום: Servlets/Customer/BuySellLoansTab. לא מצפה לקבל פרמטרים. בתשובה חוזר JSON המכיל את כל ההלוואות שהלקוח מלווה להן והוא יכול להעמידן למכירה, ובנוסף את כל הלוואות שלקוחות המערכת העמידו למכירה והוא יכול לקנות אותן.

2) 🡨 **PutLoansToSale**מיקום: Servlets/Customer/BuySellLoansTab. מצפה לקבל בגוף הבקשה JSON המכיל את שמות ההלוואות אותן המשתמש רוצה להעמיד למכירה. בתשובה חוזר פידבק לגבי הצלחת\כישלון הבקשה.

3) 🡨 **GetPaymentsInfoAndNotifications**מיקום: Servlets/Customer/PaymentsTab. לא מצפה לקבל פרמטרים. הבקשה מחזירה כJSON את כל ההלוואות שניתן לשלם עבורן תשלום בודד בי"ז הנוכחי, בין אם הן active או risk, ובנוסף מידע על כל ההלוואות שניתן לסגור כעת במכה, בין אם הן active או risk. כמו כן הבקשה מחזירה את כל התראות התשלום של הלקוח.

4) 🡨 **MakeActivePayment**מיקום: Servlets/Customer/PaymentsTab. בקשה זו מבצעת תשלום בודד של הלוואה שנמצאת במצב active והגיע מועד התשלום שלה. היא מצפה לקבל כפרמטר את שם הלוואה אותה רוצים לשלם. בתשובה חוזר פידבק לגבי הצלחת\כישלון הבקשה.

5) 🡨 **MakeRiskPayment**מיקום: Servlets/Customer/PaymentsTab. בקשה זו מבצעת תשלום בודד של הלוואה שנמצאת במצב risk והגיע מועד התשלום שלה. היא מצפה לקבל כפרמטר את שם הלוואה אותה רוצים לשלם, ובנוסף את הסכום אותו רוצים לשלם. בתשובה חוזר פידבק לגבי הצלחת\כישלון הבקשה.

5) 🡨 **MakeRiskPayment**מיקום: Servlets/Customer/PaymentsTab. בקשה זו מבצעת תשלום בודד של הלוואה שנמצאת במצב risk והגיע מועד התשלום שלה. היא מצפה לקבל כפרמטר את שם הלוואה אותה רוצים לשלם, ובנוסף את הסכום אותו רוצים לשלם. בתשובה חוזר פידבק לגבי הצלחת\כישלון הבקשה.

6) 🡨 **CloseLoan**מיקום: Servlets/Customer/PaymentsTab. בקשה זו סוגרת הלוואה במכה אחת. היא מצפה לקבל כפרמטר את שם הלוואה אותה רוצים לשלם. בתשובה חוזר פידבק לגבי הצלחת\כישלון הבקשה.

7) 🡨 **Charge Money**מיקום: Servlets/Customer/InformationTab. בקשה זו טוענת כסף לחשבון של הלקוח. מצפה לקבל כפרמטר את הסכום אותו רוצים לטעון. בתשובה חוזר פידבק לגבי הצלחת\כישלון הבקשה.

8) 🡨 **Withdraw Money**מיקום: Servlets/Customer/ InformationTab. בקשה זו מושכת כסף מהחשבון של הלקוח. מצפה לקבל כפרמטר את הסכום אותו רוצים למשוך. בתשובה חוזר פידבק לגבי הצלחת\כישלון הבקשה.

9) 🡨 **GetCustomerAndServerInformation** מיקום: Servlets/Customer/ InformationTab. בקשה זו מחזירה כJSON את כל המידעים הרלוונטיים ללקוח (כפי שמופיעים בטאב המידע באפליקציית הלקוח), הכוללים את מידעי ההלוואות שברשותו, מידעי ההלוואות שהוא מלווה להן, פעולות בחשבון, יתרה נוכחית ועוד.