

מסמר תכון המתאר את מבנה המערכת, תיאור המחלקות השונות והקשרים ביניהם

אלמוג סיטון

almogsiton@gmail.com

205501901

תוכן עניינים

פרק 1 - מבוא עמוד 3

פרק 2 - ארכיטקטורת המערכת עמודים 5-4

פרק 3 – בניית הנדרסונים עמודים 11-6

פרק 4 – תיאור המחלקות והקשרים ביניהן עמודים 14-12

פרק 5 – קבצי XHTML והקשר למחלקות עמודים 16-15

פרק 6 – זרימת מידע בין שכבות עמוד 17

פרק 7 – היבטים מערכתיים עמוד 18

פרק 1 – מבוא

1. מטרת המערכת:

מערכת **Almoviland** נבנתה לצורך ניהול ספריית סרטים דיגיטליית, ומיעודת לאפשר למשתמשים לבצע מגוון פעולות באופן מקוון, כגון: השאלת סרטים, צפייה בפרטים, ניהול ביקורות ודירוגים, והחזרת סרטים. בנוסף, ניהול הסרטים, המשتمשים, הקטגוריות, ולפקח על השאלות.

המערכת פועלת בסביבה מקומית (localhost) על שרת GlassFish.

2. מטרות הפרויקט:

- פיתוח מערכת מודולרית לניהול ספריית סרטים בשפת Java.
- שימוש עקרונות תכנון מונחה עצמים MVC.
- שימוש בטכנולוגיות Web מתקדמות כגון RESTful API, JSF, XHTML, CSS ו-JSF.
- התממשקות למסד נתונים (Apache Derby) לצורך ניהול מידע על משתמשים, סרטים, השאלות ו ביקורות.
- הבטחת חווית משתמש חלקה וידידותית תוך הפרדה בין הרשאות משתמשים.

3. טכנולוגיות וכליים:

המערכת פותחה בסביבת **Apache NetBeans 23** עם שימוש בטכנולוגיות הבאות:

- **Jakarta Java EE** פלטפורמה ליישומי Web.
- **JavaServer Faces (JSF)** תשתיית יצירה ממשיי משתמש מבוססי קומפוננטות.
- **JDBC** גישה ישירה לבסיס הנתונים.
- **XHTML + CSS + JavaScript** לבניית משך משתמש רשפניibi וдинמי.
- **RESTful Web Services (JAX-RS)** חישפת מידע בפורמט JSON.
- **GlassFish Server 7** שרת לפירסה מקומית של האפליקציה.
- **Apache Derby** מסד נתונים לניהול מידע.

פרק 2 – ארכיטקטורת המערכת

1. סקירה כללית:

מערכת Almoviland מבוססת על ארכיטקטורת (Model–View–Controller), המפרידה בין שכבת הנתונים (Model), שכבת הלוגיקה העסקית והבקשה (Controller) לשכבות התצוגה (View). מבנה זה מאפשר הפרדה ברורה בין תחומי אחראיות, תחזוקה קלה יותר של הקוד, והרחבת נוחה של רכיבים קיימים. המערכת מיועדת לפעול בסביבה מקומית, תוך שימוש בשרת GlassFish ובמסמך נתונים מסוג Apache Derby כאשר כל פעולה מתבצעת על ידי מעבר מסודר בין שכבות המערכת.

2. מודול לקוח(Client)

מודול הלקוח כולל את הממשק הגרפי של המשתמש הרגיל ומורכב מקובץ XHTML המוצבבים בעזרת CSS ונתמכים ב- JavaScript . במודול זה המשתמש יכול להירשם למערכת, להתחבר, לחפש סרטים לפי שם או קטgorיה, לשאיל סרטים, להחזיר סרטים, לדוח על אובדן, להוסיף ביקורות ולדרג סרטים. בנוסף, לכל משתמש קיים עמוד פרופיל אישי שבו ניתן לצפות בהיסטורית ההשאولات ולבצע פעולות נוספות כמו חזרה או דיווח. ממשק המשתמש מותאם לתפקידו של המשתמש המחבר ומציג רק את הרכיבים הרלוונטיים עבורו.

3. מודול ניהול(Admin)

מודול המנהל מיועד למשתמשים בעלי תפקיד ADMIN ומספק להם ממשק ניהול עשיר. באמצעותו ניתן להוסיף, לעורר ולמחוק סרטים, לנוהל קטגוריות כולל תМОנות, למחוק ביקורות של משתמשים אחרים, לנוהל את רשימת המשתמשים (כולל שינוי תפקידים ומחיקה), ולאשר דיווחי אובדן. למנהל יש גישה לעמוד ניהול ייעודי כגון moviesManagement.xhtml, categoriesManagement.xhtml, usersManagement.xhtml וlossReports.xhtml כל רכיבי הניהול מוצגים רק כאשר המשתמש המחבר הוא בעל הרשאה מתאימה.

4. מודול שרת(Server-Side Logic)

מודול השירות אחראי על עיבוד הלוגיקה העסקית של המערכת. הוא כולל מחלקות Java Beans אשר שולטות בהתקנות הדינמית של המסכים ומתווכות בין התצוגה למסד הנתונים. פעולות על הנתונים מתבצעות באמצעות מחלקות DAO (Data Access Object) אשר משתמשות ב- PreparedStatement לצורך גישה בטוחה לבסיס הנתונים. בנוסף, המערכת כוללת שירות REST מסוג RS-JAX

(MovieRestService) המאפשר שליפת נתונים בפורמט JSON דרך כתובת מוגדרת. כל החיבורים, החישובים, הווידצ'יות, והבדיקות העסקיות מנוהלים במודול זה בצורה מבוזרת, ברורה, ומודולרית.

5. שכבות הליבה ויחסי הגומלין:

המערכת מקפידה על הפרדה מוחלטת בין השכבות. שכבת ה- View בנויה בקבצי XHTML המקיימים עם שכבת ה- Beans דרך Expression Language (EL) של JSF . כל Bean מחזיק את הנתונים והפעולות עבור דף מסוים, ומתקשר עם מחלקות DAO שמבצעות את הפעולות על מסד הנתונים. שכבת ה- Model כוללת את מחלקות הנתונים כמו User, Movie , Borrowing . שכבת ה- Controller מייצגת בעיקר על ידי מחלקות Beans כמו MovieBean או UserBean אשר שולחות קריאות ומבצעות לוגיקה בהתאם לפעולות המשתמש. שירות ה- REST מהוות שכבה חיצונית נפרדת לצרכים של אינטגרציה עתידית.

פרק 3 – מבנה הנתונים
טבלת USERS

טבלה זו מכילה את כל המשתמשים הרשומים במערכת – הן משתמשים רגילים והן מנהליים.
כל שורה מייצגת משתמש ומיליה את השדות הבאים:

USERS
USER_ID מזיהה ייחודי לכל משתמש (מפתח ראשי)
NAME שם מלא של המשתמש
EMAIL כתובת הדוא"ל (יחודית)
PASSWORD הסיסמה המוצפנת של המשתמש
ROLE תפקיד המשתמש
LIMIT_BORROW_MAX מספר הסרטים המרבי שימוש יכול להושאיל
DATE_REGISTRATION תאריך ההרשמה למערכת

טבלת MOVIES

טבלה זו מייצגת את כל הסרטים במערכת.
כל שורה מייצגת סרט בודד, ומכללה את השדות הבאים:

MOVIES
MOVIE_ID מזהה ייחודי לכל סרט (מפתח ראשי)
TITLE שם הסרט
DESCRIPTION תקציר תוכן הסרט
QUANTITY סך כל העותקים שקיים למלאי
COPIES_AVAILABLE מספר העותקים הזמינים להשאלה כרגע
DATE_RELEASE תאריך יציאת הסרט
POSTER_IMAGE תמונה פויסטר בפורמט מקודד

טבלת CATEGORIES

טבלת הקטגוריות שומרת את רשימת הקטגוריות האפשרות במערכת
הטבלה כוללת:

CATEGORIES
ID_CATEGORY מזהה ייחודי לקטgorיה (מפתח ראשי)
NAME_CATEGORY שם הקטgorיה
DESCRIPTION תיאור הקטgorיה
POSTER_IMAGE תמונה ייחודית לקטgorיה בפורמט מקודד

טבלת MOVIE_CATEGORIES

זו היא טבלת קישור (join table) בין סרטים לקטgorיות.
היא מאפשרת יצירות קשר מסווג ובים לרובים: סרט יכול להשתיף למספר קטgorיות,
וKatgoria יכולה לכלול מספר סרטים. הטבלה כוללת:

MOVIE_CATEGORIES
MOVIES מפתח זר לטבלת MOVIE_ID
CATEGORIES מפתח זר לטבלת CATEGORY_ID

טבלת BORROWINGS

טבלה זו מתעדת את פעולות השאלה של סרטיים.

כל שורה מייצגת השאלה אחת ומכלול:

BORROWINGS
ID_USER
ID_MOVIE
DATE_BORROW
DATE_RETURN
ID_BORROWING
STATUS

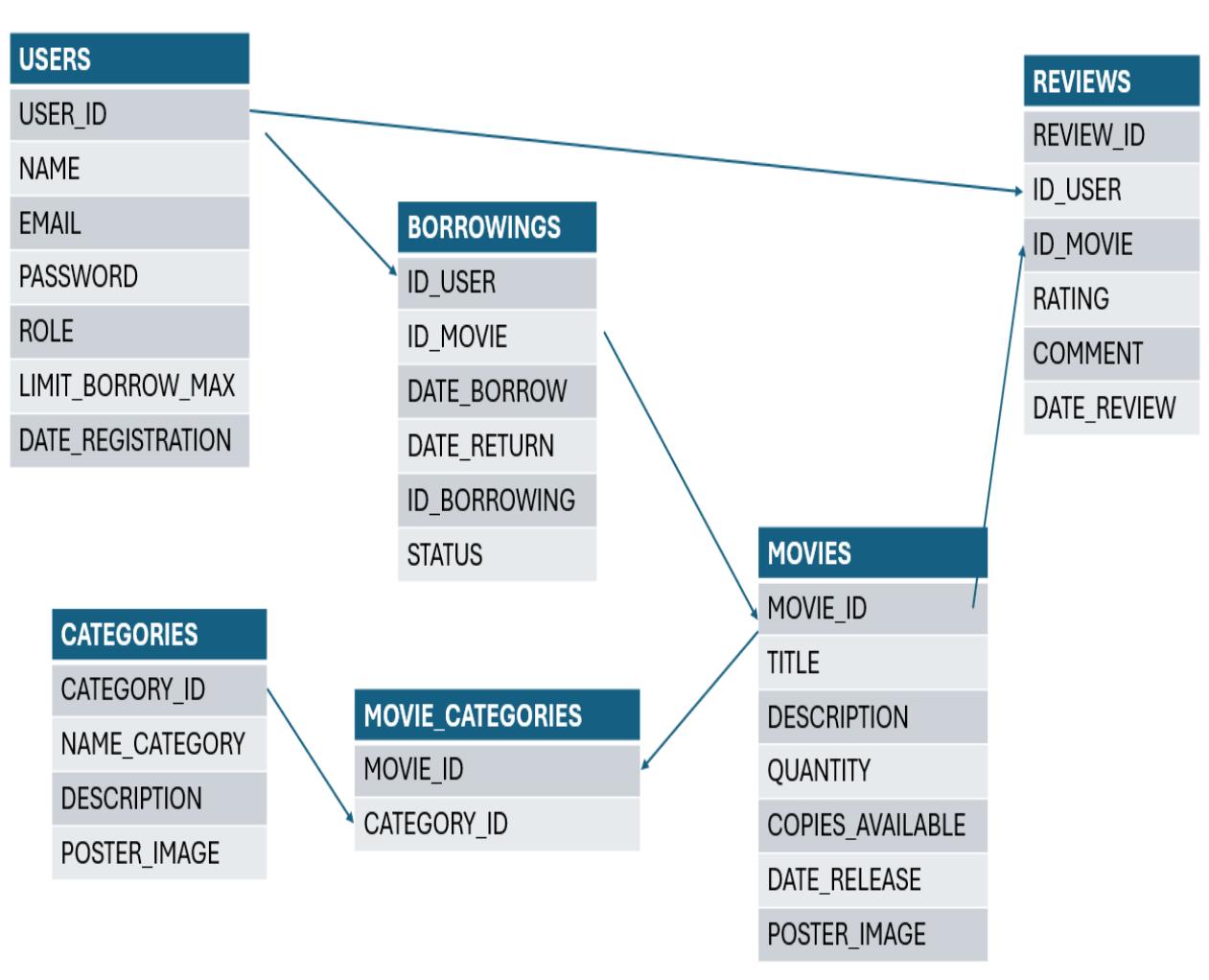
טבלת REVIEWS

בלה זו מכילה את הביקורות שנכתבו על ידי משתמשים על סרטים.
כל שורה מייצגת ביקורת אחת וכוללת:

REVIEWS
ID REVIEW מזהה ייחודי לביקורת (מפתח ראשי)
ID_USER מזהה המשתמש שכתב את הביקורת
ID_MOVIE מזהה הסרט המבוקר
RATING דירוג בין 1 ל-5
COMMENT טקסט הביקורת
DATE REVIEW תאריך כתיבת הביקורת

למערכת יש מגבלה של ביקורת אחת לכל סרט לכל משתמש, והיא נאכפת דרך הלוגיקה באפליקציה.

תיאור הקשרים במודול הנתונים



פרק 4 – תיאור המחלקות והקשרים ביניהן

מבנה כללי

- מערכת Almoviland בנויה ממספר חבילות (Packages) הממייננות את המחלקות לפי תפקידן:
- חבילת Bean אחראית על שכבות הלוגיקה והטיפול בין משק המשמש למסד הנתונים.
 - חבילת DAO כוללת את מחלקות הגישה למסד הנתונים.
 - חבילת Modules מגדירה את הישויות המיצגות את הטבלאות בסיס הנתונים (Model).
 - חבילת Utils מכילה מחלקות עזר.
 - חבילת MovieService כוללת את שירות ה-REST.
 - חבילת config מרכזת את הגדרות המערכת.

הקשרים בין המחלקות נשמרים לפי עקרון MVC כאשר כל Bean נעזר במחלקה DAO לביצוע פעולות על ישויות ה-Module ואלו מוצגות לקצה המשמש בקובצי XHTML דרך JSF.

מחלקות Bean

מחלקות ה-Bean מהוות מנהלות (Managed Beans) של JSF כל מחלקה מייצגת אזור אחר במשק המשמש או תהליך עסק, ומכליה שדות נתונים, מתודות פעולה, והפניות למחלקות DAO לדוגמה:

– **MovieBean.java** אחראית לניהול הסרטים מצד הליקוח והניהול. כוללת טעינת רשימות סרטים, סינון לפי קטגוריה, פעולות השאלה, חישוב ממוצע דירוגים, הוספה ועריכה של סרטים ועוד.

– **UserBean.java** מנהלת את המשתמש המחבר, שומרת את המידע עליו (כולל תפקיד), ומספקת מתודות לאימונות הרשות ופעולות נלוות בפרופיל.

– **BorrowBean.java** מרכזת את לוגיקת השאלה הסרטים, החזרות, דיווח על אבדן, אישורים של מנהלים.

– **ReviewBean.java** מנהלת את הוסף הביקורת, הצגת ביקורות קיימות, מחיקת ביקורת, וחישוב נתונים סטטיסטיים.

– **CategoryBean.java** מטפלת בניהול קטגוריות, כולל יצירה, עריכה, מחיקה וטעינה של תמונות.

מחלקות DAO

מחלקות ה- DAO (Data Access Object) אחראיות על התקשרות היזירה עם מסד הנתונים. כל DAO תואמת לישות אחת וכוללת שאלות SQL מוכנות לטיפול בשודות הקשורות לאוֹתָה טבלה. המחלקות עושות שימוש ב- PreparedStatement לשם מניעת SQL Injection.

– **MovieDAO.java** כולל מתודות לשילוף כל הסרטים, חיפוש לפי קטגוריה, עדכון מלאי, הוספה ועריכה של סרטים ועוד.

– **UserDAO.java** מטפל ברישום משתמשים, אימונות התחרויות, ניהול הרשאות ומעקב אחרי תפקיך המשתמש.

– **BorrowingDAO.java** מנהל השאלות, החזרות, סטטוס אובדן, ואחראי על פעולות עדכון מלאי לפי Borrowing.

– **ReviewDAO.java** מכיל לוגיקה לשילוף ביקורות לפי סרט, הוספה, מחיקה, ומינית ביקורת כפולה.

– **CategoryDAO.java** מבצע פעולות על קטגוריות ומחזיר נתונים לתצוגה ב- Bean.

מחלקות מודל (Modules)

מחלקות ה- Model הן מחלקות פשוטות שמייצגות ישות במסד הנתונים. כל מחלוקת כוללת שדות מתאימים לטבלה, בנאים, ו- getter/setter בלבד.

– **User.java** כולל שדות כמו userId, name, email, password, role, joinDate ועוד.

– **Movie.java** מייצג סרט כולל id, title, description, image, quantity ועוד.

– **Category.java** – מזאה, שם ותמונה של קטgorיה.

– **Review.java** כולל מזאה ביקורת, מזאה משתמש, מזאה סרט, דירוג, תגובה ותאריך.

– **Borrowing.java** – מזאה השאלה, מזאה משתמש, מזאה סרט, תאריכי השאלה והחזרה, וסטטוסו .

שירותי REST

במערכת קיימת שירות REST בשם **MovieRestService.java**, המחזיר את רשימה
הסրטיהם בפורמט JSON דרך הכתובת:
<http://localhost:8080/almoviland/api/movies>.
השירות מנוהל ע"י המחלקה **RestApplication.java** שגדירה את הכתובת הבסיסי
השירות מבוסס על JSR-RS-JAX ומאפשר שליפה מהירה של נתונים עבור ליקוחות חיצוניים.

מחלקות עזר ו-Config

– **AppConfig.java** מרכזת הגדרות קבועות של המערכת, כולל כתובות מסד הנתונים,
משתמש בירית מיחל, גבול השאלות ועוד.
– **PageController.java** מנהלת את ניווט הדפים במערכת לפי משתנה `currentPage`.
– **DateUtils.java, MathUtils.java** –
מחלקות סטטיות לפעולות חישוב ותאריך.
– **PasswordMatchValidator.java** –
מחלקה ולידציה מותאמת לשABL כדי לבדוק התאמה בין סיסמה לאימות סיסמה בעת רישום.

פרק 5 – קבצי XHTML והקשר למחולקות

home.xhtml - מציג את עמוד הבית עם רשימות של סרטיים פופולריים וסרטים לפי קטגוריה. משתמש ב- movieBean וב- categoryBean כדי להציג סרטים ולמנוע בין עמודים באמצעות AJAX.

movieDetails.xhtml - מציג מידע מלא על סרט שנבחר, כולל תמונה, תיאור, תאריך יציאה, כמות זמינה, דירוג ממוצע, וכפתורים לפעולה (השאלה, הוספה ביקורת, צפיה בביקורות). מבוסס על : movieBean, reviewBean borrowBean .

addReview.xhtml | viewReviews.xhtml - מאפשרים למשתמשים רשותם להווסף ביקורת על סרט או לצפות בביקורות קיימות. ב viewReviews.xhtml קיימת הרשות מחיקה לבעל הביקורת ולמנהלה. שניהם פועלם דרך reviewBean .

profile.xhtml - מאפשר למשתמש לצפות בהשאלות הפעילות שלו, להחזיר סרטים, לדוח על אובדן, ולצפות בהיסטוריה. מחובר ל userBean , borrowBean, movieBean .

register.xhtml | login.xhtml - עמודים שמנחים תהליך ההרשמה וההתחברות של המשתמש, כולל לידציה של סיסמה, אימות אשראי, הגבלת אורך סיסמה, ודפו אימייל. מבוססים על userBean .

adminDashboard.xhtml - מכיל קישורים עיקריים עבור מנהל: ניהול סרטים, קטגוריות, משתמשים, ובקשות לאובדן. לא כולל לוגיקה עסקית אלא רק ניווט.

moviesManagement.xhtml | editMovie.xhtml | editCategory.xhtml | categoriesManagement.xhtml - עמודים המאפשרים למנהל להווסף סרטים חדשים, לעורר פרטי סרט קיימים, ולבצע פעולות על המלאי. משתמשים בחאנט Bean .

editCategory.xhtml | categoriesManagement.xhtml - עמודים המאפשרים הוספה, עריכה ומחיקה של קטגוריות כולל תמונה, דרך categoryBean .

usersManagement.xhtml | userBorrowingHistory.xhtml | usersManagement.xhtml - עמודים לניהול המשתמשים – צפיה במידע, שינוי תפקידים, מחיקה, וצפיה בהיסטורית ההשאלות. מבוססים על userBean .

reportLoss.xhtml | adminLossRequests.xhtml משמש רגיל מדוח על אובדן סרט עם פרטי אשראי, ומנהל מאשר את האובדן דרך טבלת אישורים. מחובר ל borrowBean .

search.xhtml - משיק חיפוש דינמי עם תצוגת תוצאות מיידית לפי טקסט שהוזן. נתמך על ידי keyup.movieBean וככלAJAX עם .

קומפוננטות משותפות והרכבת הדפים

מערכת Almoviland עשויה שימוש בקובץ index.xhtml כשלד הראשי של הממשק. בתוך קובץ זה נטען עםוד התוכן באופן דינמי לפי ערך משתנה currentPage שב-`<h:panelGroup, PageController, ui:include>`.

בראש כל עמוד נתען באופן קבוע קובץ header.xhtml, שמכיל את תפריט הניווט הראשי. תפריט זה מותאם להרשאות המשתמש:

אם המשתמש אינו מחובר – מוצגים קישורים ל Register ו Login .

אם המשתמש מחובר מוצגים שמו כפתור Logout או Profile .

אם המשתמש הוא Admin מוצג גם קישור ל Admin Dashboard .

בנוסף לכך, הודעות מערכת (שגיאות, אישור, מידע) מוצגות בכל עמוד על ידי רכיב `<h:messages>` אשר נתוע באופן ישיר בכל קובץ שבו יש צורך בהצגת תגובה.

כל הדפים שומרים על סגנון אחד בעזרת קובץ CSS מרכזי styles.css וקובץ JavaScript scripts . המוטמעים דרך `<h:outputStylesheet>` ו `<h:outputScript>` .

כך נוצר ממשק אחיד, רספונסיבי, המופעל כולו מתוך שלד אחד – ומוצגים בו תכנים משתנים באופן מודולרי.

סיכום

המבנה של pageController משתמש במערכת טענה דינמית של רכיבים לפי , pageController , בכך מאפשר שליטה קלה על ניווט, הצגת מסכים לפי הרשאות, והפרדה בין תפקידים.

כל XHTML מקשור ישירות למחלקה Bean אחת לפחות , וכל פעולה בממשק מעודכנת בזמן אמת בצד שרת דרך AJAX .

פרק 6 – זרימת מידע בין שכבות

מערכת Almoviland פועלת לפי עקרונות MVC (Model–View–Controller) כאשר הנתונים נעים בין שכבת התצוגה (JAXX) , שכבת הלוגיקה (Beans) ושכבת הגישה לננתונים (DAO). Zarimut ha-midu' mbooset ul shi'mosh b-SF, kol tagiot binding l'shadot b-Managed Beans. ועבודה um AJAX le-udcon dinami shel tocon b-daf lala re'enu molaa.

casar meshatmesh mabatz peulah b-meshuk, ldogma: lochz ul cftor "hshel srt", peulah mouveret dror cftor <f:ajax> h:commandButton או <f:ajax>. cftor zeh mafuil matoda matot Bean matais.

matoda b-Bean achraiyt libdok tana'im (cmo megalat ha-shalot ao zmionot ha-uotkim) volach makan koreat lmachlik DAO aser mbaatzat at udcon b-mesad ha-nanotim.

hi DAO machzir at tzachat ha-bitzou v-ha-bean b-hatams mafuil logika uskiyit mshlima: tzagat hodua lemashatmesh, re'enuon nanotim, ao shinui daf.

am yish zoruk, hi Bean yudcon gam shadot acharim (cmo cmot zmina b-srt ao historiyit ha-shalot be-phefil ha-mashatmesh).

b-mekabil, ha-nanotim shmozgim ul ha-mskr (cmo rishimat ha-srtim ao prti ha-mashatmesh) n-kraim matot hi Bean omouverim ltzoga ba-amatzut EL.

shadot batposim cmo comment, rating, email v-ko' mkoashrim l'shadot b-, Bean b-kr shashinim shmbatz ha-mashatmesh mouverim otomotit lshefet dror ha-mengenon shel JSF.

meurut ha-hodutot <h:messages> maziga tagbotot mahaferet (lmashl: hzla, shgia, hatarah), b-hatam l-fuilot shbazu.

cmo kn, nusa she'imosh b-<f:ajax> lhazig tzachot chifush b-zman amta, le-udcon chlikim b-daf (cmo rishimat srtim l-pi katgoria ao tzachot chifush), v-kr nshmerat choviyut meshatmesh chlka v-intraktivit.

Zarimut zo chozret ul uzma ba-kl peulah b-me'arot - risom, hthborot, ha-shala, ha-zura, dirrog, niyhol v-oud - v-kl ricib b-me'arot mmash at chlko l-pi tefkidot: View mazig Bean matot DAO niges l-mesad.

פרק 7 – היבטים מערכתיים

אבטחת מידע

המערכת כוללת מספר מנגנוני אבטחה בסיסיים לשמירה על פרטיות המשתמשים ושלמות המידע. סיסמאות המשתמשים נשמרות במסד הנתונים לאחר הצפנה באמצעות אלגוריתם Hash מסוג SHA-256, כך שלא ניתן לקרוא אותן בצורה גלויה. כל הפעולות מול מסד הנתונים מבוצעות באמצעות PreparedStatement דבר שמנוע התקפות מסוג SQL Injection.

כמו כן, למשתמשים יש גישה רק לאזרחי המערכת המותאמים לתפקידם – משתמש רגיל לא יכול להיכנס לעמוד ניהול, והלוגיקה בJSF שומרת על כך גם מצד התצוגה וגם מצד השרת.

בקרת הרשותות ותפקידו משתמש

במערכת קיימים שני תפקידים משתמש: ADMIN ו USER. ההרשות נקבעת בעת ההרשמה או על ידי שינוי דרך משקל ניהול. ההרשות מושפעה על התפריטים המוצגים, הדפים הזמינים, והפעולות שנitin לביצוע (כמו מחיקת משתמשים, ניהול מלאי, אישור אובדן ועוד). כל עמוד XHTML כולל תנאי rendered שמנוע הצגת רכיבים למשתמשים שאינם בעלי ההרשות המתאימה, ובצד החאנט Bean קיימת לוגיקה שמבצעת בדיקות חוזרות גם על הפעולות עצמן.

ולידציה ובקרת קלט

המערכת מבצעת ולידציה (אימונות תקינות קלט) הן מצד הלוקוח והן מצד השרת. לדוגמה: בבדיקה אימייל, התאמת סיסמה לאימונות סיסמה, תקינות של כרטיס אשראי (אורך תוקף ו-CVC).

בדיקות אלו נעשות באמצעות תבניות pattern, הודיעות שגיאה מותאמות, ולוגיקת אימונות פנימית בחאנט Bean. בשודות כמו comment ו rating נעשה ולידציה למניעת הגשה לא תקינה של ביקורות.

מניעת התנגשויות בשאלת סרטיים

כאשר משתמש מנסה להשאיל סרט, המערכת מפעילה שאילתת SQL שמורידה עותק מהכמתות הכוללת רק אם קיימים עותקים זמינים גדולות מ 0.

העדכן מתבצע בפועל אחת – כך שגם אם שני משתמשים לוחצים על "השאיל" במקביל, רק אחד יצליח. המערכת מונעת בכך השאילת יתר ושומרת על עקביות מלאה במלאי.