**LICENSEBOX**

**אפיון המערכת**

‏2013

**פרויקט גמר בתכנות מתקדם בשפת Java**

**אנה גוזמן, מיכאל פלצב**

תוכן עניינים

[תיאור כללי 2](#_Toc367133122)

[סוגי משתמשים במערכת (קבוצות הרשאה) 2](#_Toc367133123)

[דוחות 3](#_Toc367133124)

[אופן הניהול של רישיונות במערכת 3](#_Toc367133125)

[תהליך אישור רכישה/שדרוג 4](#_Toc367133126)

[טכנולוגיות בשימוש המערכת 5](#_Toc367133127)

[כניסה למערכת 5](#_Toc367133128)

[בסיס הנתונים 5](#_Toc367133129)

[ERD 7](#_Toc367133130)

[ארכיטקטורת המערכת 8](#_Toc367133131)

[שכבת ניהול הנתונים – com.licensebox.db 8](#_Toc367133132)

[חבילה com.licensebox.db.entity 8](#_Toc367133133)

[חבילה com.licensebox.db.dao 8](#_Toc367133134)

[שכבת הלוגיקה העסקית – com.licensebox.bl 11](#_Toc367133135)

[חבילה com.licensebox.bl.email 11](#_Toc367133136)

[חבילה com.licensebox.bl.exception 11](#_Toc367133137)

[חבילה com.licensebox.bl.license 11](#_Toc367133138)

[חבילה com.licensebox.bl.requests 12](#_Toc367133139)

[שכבת ממשק המשתמש – com.licensebox.ui 13](#_Toc367133140)

[התחברות למערכת 13](#_Toc367133141)

[ממשק משתמש כללי 13](#_Toc367133142)

[ממשק משתמש למנהלן 14](#_Toc367133143)

[ממשק משתמש למנהל הרשיונות 14](#_Toc367133144)

[תפריט למנהל 14](#_Toc367133145)

[תפריט לראש צוות 15](#_Toc367133146)

[חבילה com.licensebox.ui.session 15](#_Toc367133147)

[חבילה com.licensebox.ui 15](#_Toc367133148)

[חבילה com.licensebox.ui.converter 15](#_Toc367133149)

# תיאור כללי

בסביבה המתקדמת שבה אנו חיים, תופסות מערכות מחשב חלק מאוד חשוב בחיינו. חברות רבות מחזיקות מערכות מחשב מתקדמות אשר עוזרות להן לנהל את עסקיהן.

אך לא די בחומרה עצמה ועל חברות אלה להצטייד בנוסף גם בתוכנות מתאימות שעלותן נעמדת במאות שקלים ואף באלפי שקלים עבור כל רישיון.

לרוב, כאשר נרכשת תוכנה, נרכש בעצם רישיון שימוש בתוכנה (Serial Number) והוא מוקצה עבור מחשב מסוים או לחלופין עבור משתמש מסוים.

וכך, ככל שהחברה גדלה, מנהליה מוצאים את עצמם מוציאים יותר ויותר עבור תוכנות מבלי שהם מצליחים לנהל את הרישומים עבור הרכישות האלה.

בעקבות זאת, הגיעה דרישה למערכת שתוכל לנהל את רישיונות התוכנה בעסק.

מסמך זה בא לתאר את מערכת ה"LicenseBox" שנותנת מענה לבעיה המוצגת לעיל.

# סוגי משתמשים במערכת (קבוצות הרשאה)

**משתמש (User)** – משתמש שהוא מפתח רגיל, שמורשה לבצע את הפעולות הבאות:

* להתחבר למערכת
* לצפות ברשימת הרישיונות שניתנו לו
* לבקש רישיון חדש לתוכנה
* להחזיר רישיון שאין לו צורך בו

**ראש צוות (Team Leader)** – ראש צוות מפתחים, שמורשה לבצע את הפעולות הבאות (בנוסף לפעולות של משתמש רגיל):

* לאשר לחברי הצוות שלו את הבקשות לרשיונות חדשים
* לראות כמות רישיונות לכל צוות

**מנהל רשיונות (License Manager)** – משתמש שאחראי על ניהול כל הרשיונות בחברה. ה-License Manager יכול להיות כל אחד מהמשתמשים. מורשה לבצע במערכת את הפעולות הבאות (בנוסף ליכולות של משתמש רגיל):

* לראות את כל הרישיונות של כל הצוותים
* להוסיף רישיון חדש לתוכנה קיימת
* לעדכן רישיונות קיימים
* להחזיר רישיון של משתמש למאגר
* לראות נתונים מספריים על רכישות, כמות רישיונות לכל תוכנה ונתונים עבור כל צוות

**מנהל (Manager)** – משתמש שאחראי על רכש של תוכנות ורשיונות חדשים עבור החברה. יכול לבצע את הפעולות הבאות:

* לראות נתונים לגבי רכישות של רישיונות ונתונים לגבי רישיונות אצל צוותים
* לאשר הוספת רישיונות חדשים למאגר (הבקשות מגיעות ממנהל הרישיונות)

**מינהלן (Administrator)** – משתמש מיוחד בשם admin שאחראי לבצע את הפעולות הבאות:

* להוסיף משתמשים חדשים למערכת
* להוסיף קבוצות חדשות
* לשייך משתמשים שונים לקבוצות שונות ולסוגי משתמשים שונים

בנוסף, מנהלן יוכל לראות לוג של פעילות שמתבצעת במערכת, כגון התחברות של משתמשים, הגשת בקשות לאישור וכ"ד.

## דוחות

לכל סוג משתמש (חוץ ממשתמש הפשוט) קיימים במערכת דוחות תקופתיים שמאפשרים לראות את מספר הרישיונות ותוכניות החדשות לתקופה מסוימת, הרחבת הצוותים וכ"ד.

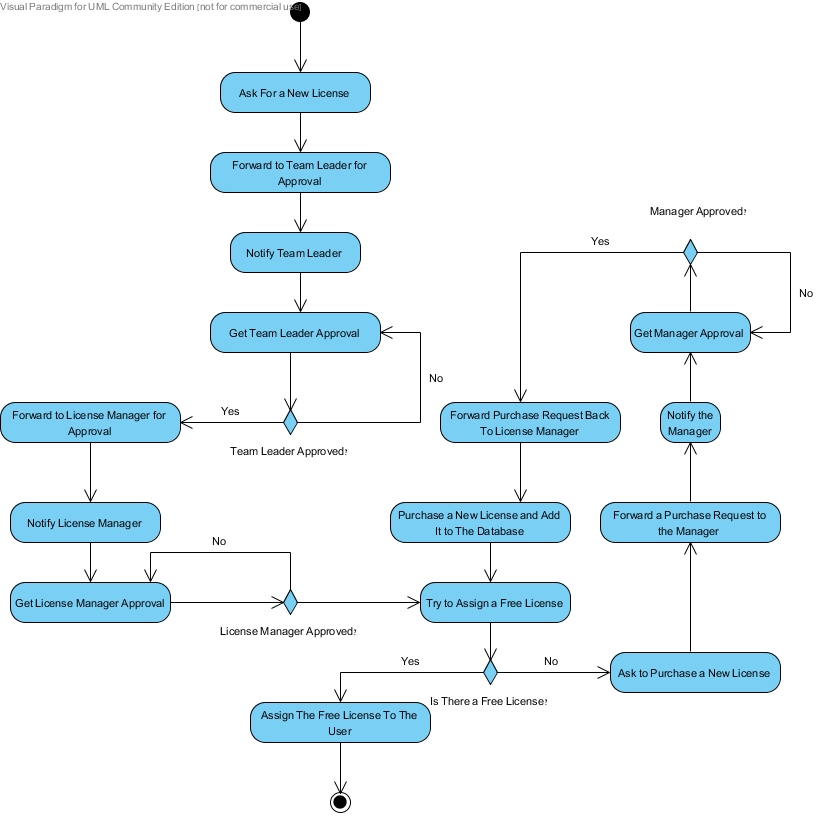
# אופן הניהול של רישיונות במערכת

כאשר רישיון משויך למשתמש מסוים, הוא מסומן כ"תפוס" ולא ניתן לשייך אותו למשתמש נוסף. ברגע שמשתמש מסיים את השימוש ומחזיר את הרישיון, הרישיון מתווסף חזרה למאגר של רישיונות פנויים. כל ההיסטוריה של הרישיון תשמר.

כאשר משתמש יוצר בקשה חדשה לרישיון, הבקשה עוברת אוטומטית לאישור ראש הצוות שלו (ראש הצוות מקבל עדכון במייל). לאחר אישור ע"י ראש הצוות, הבקשה עוברת לאישור מנהל הרישיונות (גם הוא מקבל עדכון במייל).

במידה ויש רישיון פנוי לתוכנה המבוקשת, מנהל ההרשאות מאשר את הבקשה ומשייך את הרישיון למשתמש. במידה ואין רישיון פנוי, מנהל הרישיונות יוצר בקשה לרכישת רישיון חדש שעוברת לאישור המנהל.

## תהליך אישור רכישה/שדרוג



# טכנולוגיות בשימוש המערכת

מערכת "LicenseBox" משלבת בתוכה מספר טכנולוגיות:

* MySQL Server – מסד נתונים של המערכת
* JPA (Java Persistence API) – ספריית Java מיוחדת שמאפשרת למפות אובייקטים ב Java לבסיס נתונים לרציונלי
* JPQL – ל JPA-יש שפת שאילתות שנקראת JPQL והיא דומה מאוד ל SQL-
* EJB (Enterprise JavaBeans) – ארכיטקטורה לבנייה מודולרית של יישומי enterprise. בכתיבת המערכת נעשה שימוש ב-JavaBeans הבאים:
  + SessionBean
* JSF 2.2 (Java Server Faces) – טכנולוגיה המשמשת לבניית ממשק משתמש. בפרויקט נעשה שימוש ב-ManagedBeans למימוש רכיב ה-controller
* Facelets – קבצי xhtml עם קוד EL למימוש רכיב ה-view
* AJAX – הרצת קוד להצגת נתונים בצד לקוח במקום בצד שרת
* CSS – Cascading Style Sheets לעיצוב דפי המערכת
* PrimeFaces – ספריה של רכיבים מתקדמים לבניית ממשק משתמש, לשימוש עם Facelets
* Java Mail – חבילה של Java המכילה מחלקות עזר לשליחת מיילים בעזרת קוד של Java
* Log4j (Apache Log4j API) – רכיב המשמש ליצירת לוגים במערכת

# כניסה למערכת

הגישה למערכת מתבצעת ב-HTTP לשרת GlassFish שמאזין בפורט 8080.

לינק: <http://localhost:8080/LisenceBox-web/login.xhtml>

לינק זה מנתב את המשתמש למסך הזדהות שבו חייבים להזין שם משתמש וסיסמא. לאחר הזנת שם משתמש וסיסמא, כל משתמש יקבל תפריט לפי ההרשאות שקיימות עבורו במערכת (לפי סוג משתמש שאליו הוא משויך).

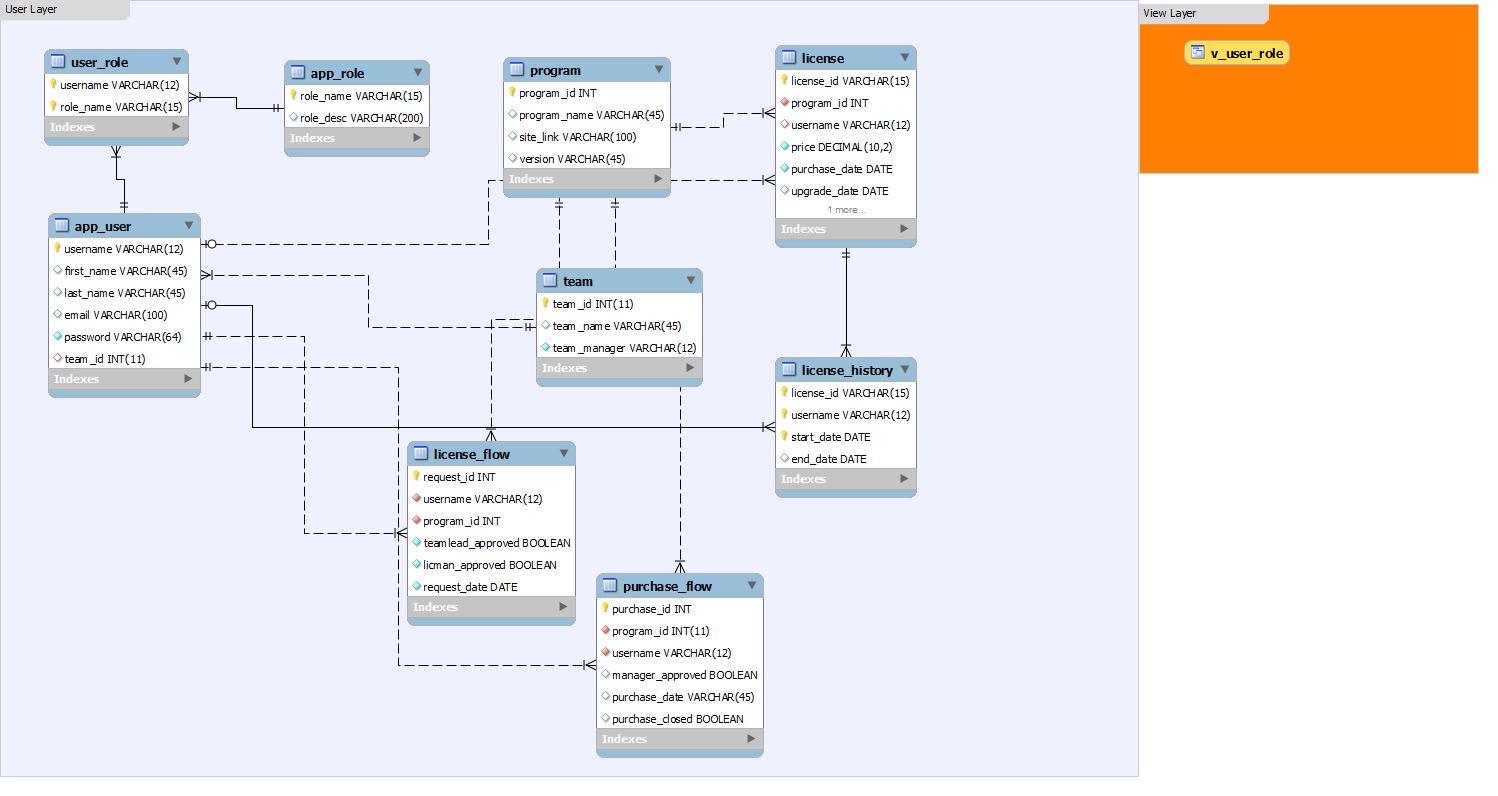
ניהול הגישה והאבטחה לאפליקציה נעשה ע"י מערכת מובנית של שרת ה-Glassfish בעזרת הגדרות שנעשו באפליקציה עצמה ובנתונים מתוך מסד הנתונים בו שמורים כל המשתמשים וקבוצות ההרשאה אליהן הם שייכים.

# בסיס הנתונים

בסיס הנתונים נקרא licenseboxdb ומכיל את הטבלאות הבאות:

|  |  |  |
| --- | --- | --- |
| **שם הטבלה** | **שדות בטבלה** | **הגדרת הטבלה** |
| app\_user | username  first\_name  last\_name  email  password  team\_id | הטבלה מכילה את כל המשתמשים במערכת ואת פרטיהם, כגון שם פרטי ומשפחה, מייל, סיסמא ו-ID של צוות שאליו שייך המשתמש (יכול להיות מצב שמשתמש לא שייך לאף צוות).  מפתח ראשי: username  מפתח זר: team\_id |
| app\_role | role\_name  role\_desc | הטבלה מכילה את הקבוצות המוגדרות במערכת. הרשאות היוזר נקבעות לפי הקבוצה שהוא משויך אליה, וסה"כ יש 5 קבוצות: USER, TEAM\_LEADER, LICENSE\_MANAGER, MANAGER, ADMIN  מפתח ראשי: role\_name |
| user\_role | username  role\_name | טבלה שמקשרת בין משתמש לקבוצה.  מפתח ראשי: username,role\_name. |
| program | program\_id  program\_name  site\_link  Version | טבלה שמכילה את כל התוכנות שארגון כבר רכש. לכל תוכנה יש זיהוי חח"ע שלה.  מפתח ראשי: program\_id |
| license | license\_id  program\_id  username  price  purchase\_date  upgrade\_date  attachment | טבלה שמכילה את כל הרשיונות שקיימים לארגון ופרטים הבאים עבור כל רשיון: ID של הרשיון, עבור איזה תוכנה הרשיון, לאיזה יוזר משויך (אם השדה ריק המשמעות היא שהרשיון פנוי), מחיר, תאריך רכשיה ועדכון, וקובץ מצורף של התכתבות במייל בנוגע לרכישת רישיון).  מפתח ראשי: license\_id  מפתחות זרים: program\_id ו-username |
| team | team\_id  team\_name  team\_manager | הטבלה מכילה את הצוותים שיש בארגון ומי הראש צוות.  מפתח ראשי: team\_id |
| license\_history | license\_id  username  start\_date  end\_date | טבלה ששומרת היסטוריה של כל שיוך רשיון למשתמש. ברגע שמשתמש מחזיר את הרשיון מתעדכן תאריך הסיום.  מפתח ראשי: license\_id, username, start\_date |
| license\_flow | request\_id  username  program\_d  teamlead\_approved  licman\_approved  request\_date | טבלת בקשות להקצאת רשיון ליוזר. כל שורה מכילה מספר הבקשה (מספר רץ), היוזר המבקש, התוכנה שעבורה נדרש הרשיון, שני flagים שמציינים האם הבקשה אושרה ותאריך הבקשה.  מפתח ראשי: request\_id  מפתחות זרים: username ו-program\_id |
| purchase\_flow | purchase\_id  program\_id  username  manager\_approved  purchase\_date  purchase\_closed | טבלת בקשות לרכישת רשיון חדש. כל שורה מכילה מספר הבקשה (מספר רץ), היוזר המבקש, התוכנה שעבורה נדרש הרשיון, flag שמציין האם הבקשה אושרה ותאריך הבקשה.  מפתח ראשי: purchase\_id  מפתחות זרים: username ו-program\_id |

## ERD



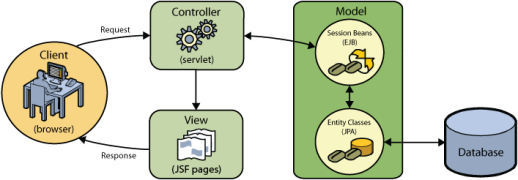
# ארכיטקטורת המערכת

מערכת "LicenseBox" פותחה בארכיטקטורת שלוש השכבות, כך שכל מחלקות ה-JAVA מחולקות ל-3 חבילות:

* com.licensebox.db
* com.licensebox.bl
* com.licensebox.ui

שתי החבילות הראשונות נארזות בקובץ jar ואילו החבילה השלישית (ui) נארזת בקובץ war. לאחר מכן, שתי החבילות נארזות בקובץ אחד בשם ear והוא זה שנפרס לשרת.

סכמת המערכת:



## שכבת ניהול הנתונים – com.licensebox.db

שכבת ניהול הנתונים אחראית על הגישה ל-DB ומיוצגת בשתי חבילות:

### חבילה com.licensebox.db.entity

חבילה זו מכילה entities שכל אחד מהם מייצג טבלה ב-DB. עבור כל Entity קיימים NamedQueries לשליפה מהטבלה לפי כל שדה, ו-setters ו-getters לכל פרמטר של מחלקה שמייצג עמודה בטבלה.

מחלקות של business logic ו- user interface לא אמורות לגשת ל-DB דרך ה-entities. הגישה ל-DB מתבצעת בעזרת Data Access Objects (DAOs) שמפורטים למטה.

כל entity ממש את הממשק Serializable.

### חבילה com.licensebox.db.dao

DAO זהdesign pattern שמטרתו ליצור Interface אחיד ל-DB כך שלשכבת האפליקציה יהיו שקופים כל השינויים בשכבת ה-DB (לדוגמא, מעבר מ-MySQL ל-Oracle). במערכת LicenseBox קיים interface ייעודי לכל Entity:

1. AppRoleDaoLocal - ממשק לטבלה app\_role דרך Entity AppRole:

|  |  |
| --- | --- |
| מתודה | תאור |
| updateRole | מעדכנת קבוצה בטבלה |
| findRole | מחזירה שורה בטלבה |

1. AppUserDaoLocal – ממשק לטבלה app\_user דרך Entity AppUser:

|  |  |
| --- | --- |
| מתודה | תאור |
| getUserRoles | מחזירה רשימת קבוצות שאליהן משויך היוזר |
| getUserByRolename | מחזירה רשימת משתמשים שמשויכים לקבוצה מסוימת |
| getUserFullName | מחזירה מחרוזת שמייצג שם פרטי ושם משפחה של משתמש |
| getUserList | מחזירה את כל המשתמשים במערכת |
| getUserByUsername | מחזירה פרטים של משתמש מסוים |
| updateUser | מעדכנת פרטי משתמש |
| createNewUser | יוצרת משתמש חדש |
| getUserTeamName | מחזירה שם של צוות של המשתמש |
| setUserPassword | משנה סיסמא של משתמש (שמורה בטבלה בצורה מוצפנת) |

1. LicenseDaoLocal – ממשק לטבלת license דרך License Entity:

|  |  |
| --- | --- |
| מתודה | תאור |
| createNewLicense | יוצרת רשיון חדש בטבלה |
| updateLicense | מעדכת רשיון קיים |
| getLicenseById | מחזירה רשיון לפי ID מסוים |
| getLicensesByProgramId | מחזירה את כל הרשיונות השייכם לתוכנה מסוימת |
| getFreeLicensesByProgram | מחזירה את כל הרשיונות הפנויים עבור תוכנה מסוימת |
| getAll | מחזירה את כל הרשיונות במערכת |
| getLicensesByAppUser | מחזירה את כל הרשיונות המשויכים למשתמש מסוים |

1. LicenseFlowDaoLocal – ממשק לטבלת license\_flow דרך LicenseFlow Entity:

|  |  |
| --- | --- |
| מתודה | תאור |
| createRequest | יוצרת בקשה לרשיון חדש ומחזירה את מספר הבקשה |
| editRequest | מעדכנת בקשה קיימת |
| deleteRequest | מוחקת בקשה |
| getByRequestId | מחזירה בקשה לפי מספר הבקשה |
| getByUsername | מחזירה כל הבקשות של משתמש מסוים |
| getForTeamleadApproval | מחזירה את כל הבקשות שמחכות לאישור של ראש צוות מסוים |
| getForLicmanApproval | מחזירה את כל הבקשות שמחכות לאישור מנהל הרשיונות |
| getTeamLeadForRequest | מחזירה את שם משתמש של ראש הצוות שאמור לאשר בקשה מסוימת |

1. LicenseHistoryDaoLocal – ממשק לטבלת license\_history דרך LicenseFlow Entity

|  |  |
| --- | --- |
| מתודה | תאור |
| createHistoryEntry | רושמת שורה חדשה של הסיטוריית שיוך של רשיון למשתמש |
| editHistoryEntry | מעדכנת היסטוריה של רשיון |
| getLicenseHistory | מחזירה היסטוריה של רשיון מסוים |
| getLicenseHistoryByUsername | מחזירה הסטוריית רשיונות עבור משתמש מסוים |

1. ProgramDaoLocal – ממשק לטבלת program דרך Program Entity:

|  |  |
| --- | --- |
| מתודה | תאור |
| createProgram | מוסיפה תוכנית חדשה למאגר |
| editProgram | מעדכנת פרטי תוכנית קיימת |
| getByProgramId | מחזירה פרטי תוכנית לפי ID |
| getAll | מחזירה את כל התוכנות הקיימות במאגר |

1. PurchaseFlowDaoLocal – ממשק לטבלת purchase\_flow דרך PurchaseFlow Entity

|  |  |
| --- | --- |
| מתודה | תאור |
| createPurchaseFlow | יוצרת בקשה לרכישת רשיון חדש ומחזירה מספר הבקשה |
| editPurchaseFlow | מעדכנת בקשה קיימת לרכישת רשיון |
| getPurchaseFlow | מחזירה פרטי הבקה לפי מספרה |
| getAll | מחזירה את כל הבקשות לרכישת הרשיונות במערכת |
| getForManagerApproval | מחזירה את כל הבקשות לרכישת רשיונות שממתינות לאישור מנהל |

1. TeamDaoLocal – ממשק לטבלת team דרך Team Entity

|  |  |
| --- | --- |
| מתודה | תאור |
| createNewTeam | מוסיפה צוות חדש |
| updateTeam | מעדכנת פרטי הצוות |
| removeTeam | מוחקת צוות |
| getTeamById | מחזירה פרטי צוות לפי ה-ID שלו |
| getTeamByName | מחזירה פרטי צוות לפי השם שלו |
| getTeamsByTeamManager | מחזירה את כל הצוותים שמנהל משתמש מסוים |

## שכבת הלוגיקה העסקית – com.licensebox.bl

השכבה אחראית על הפונקציונליות של המערכת ומורכבת מחבילות הבאות:

* com.licensebox.bl.email
* com.licensebox.bl.exception
* com.licensebox.bl.license
* com.licensebox.bl.requests

### חבילה com.licensebox.bl.email

חבילה זו מספקת ממשק לשליחת מיילים למשתמשי המערכת. המיילים מעדכנים את המשתמשים בפעולות שונות, כגון שינוי סיסמא או בקשה חדשה שמחכה לאישור.

ממשק LicenseBoxEmailLocal:

|  |  |
| --- | --- |
| מתודה | תיאור |
| sendMessage | המתודה שולחת מייל למשתמש כלשהו מקבלת ארבעה פרמטרים: השם המלא של הנמען, נושא המייל, תוכן המייל וכתובת הדואר האלקטרוני של הנמען. |

את הממשק מממשת כעת מחלקה אחת המוגדרת לעבוד עם תיבות מען של Gmail. הכתובת שמוגדרת בו כרגע היא [licenseboxproject@gmail.com](mailto:licenseboxproject@gmail.com) וממנה נשלחים כל המיילים במערכת.

### חבילה com.licensebox.bl.exception

בחבילה זו ישנה מחלקה אחת בשם LicenseBoxUncheckedException שמרחיבה את המחלקה EJBException. המחלקה הינה בעלת בנאי יחיד המקבל כפרמטר משתנה מסוג String שהוא בעצם הודעת השגיאה.

### חבילה com.licensebox.bl.license

חבילה זו מטפלת ב-flow של בקשות חדשות לשיוך רשיונות קיימים או רכישת רשיונות חדשים.

ממשק LicenseManagerLocal:

|  |  |
| --- | --- |
| מתודה | תאור |
| createNewLicenseRequest | מקבלת כפרמטר שם משתמש המבקש ו-ID של תוכנה שעבורה נדרש הרשיון, יוצרת בקשה חדשה, שולחת עדכון במייל לראש הצוות של המשתמש, ומחזירה את מספר הבקשה |
| approveByTeamLeader | מקבלת את מספר הבקשה ושם הראש צוות, מסמנת את הבקשה כמאושרת ע"י ראש הצוות ומעבירה לאישור מנהל הרשיונות (העדכון נשלח במייל) |
| approveByLicenseManager | מקבלת את מספר הבקשה, מסמנת אותה כמאושרת ומשייכת את הרשיון למשתמש המבקש. |
| logLicenseHistory | רושמת את השיוך למשתמש חדש לטבלת ההיסטוריה |
| addAttachment | מקבלת את פרטי הרשיון ונתיב לקובץ, ומצרפת את הקובץ לרשיון |
| getAttachment | מחזירה קובץ שמצורף לרשיון מסוים ושומרת אותו בנתיב שמועבר כפרמטר |
| assignFreeLicense | מקבלת מספר בקשה לרשיון ובמידה והיא מאושרת משייכת את הרשיון למשתמש המבקש |
| returnLicense | מחזירה את הרשיון למאגר הרשיונות הפנויים |
| createNewPurchaseRequest | מקבלת כפרמטר שם משתמש המבקש ו- ID של תוכנה שעבורה נדרש הרשיון, יוצרת בקשה חדשה לרכש רשיון חדש, שולחת עדכון במייל למנהל, ומחזירה את מספר הבקשה |
| approveByManager | מקבלת את מספר הבקשה לרכש רשיון חדש ומסמנת אותה כמאושרת ע"י המנהל |
| purchaseRequestCompleted | מסיימת את הרכישת הרשיון החדש |

### חבילה com.licensebox.bl.requests

חבילה זו מרכזת בקשות ליצירת משתמשים וצוותים חדשים (פעולות שמבוצעות ע"י מנהלן בלבד)

CreateUserRequestLocal

|  |  |
| --- | --- |
| מתודה | תאור |
| setNewPasswordForExistingUser | מאפסת סיסמא למשתמש קיים. הסיסמא הינה רנדומלית ונשמרת בטבלה בצורה מוצפנת. משתמש מקבל מייל עם סיסמתו החדשה. |
| createNewUser | יוצרת משתמש חדש במערכת. משתמש מקבל מייל עם פרטי ההתחברות למערכת. |
| setNewManager | משייכת משתמש נתון לקבוצת מנהל (MANAGER). |
| removeManager | מורידה ממשתמש נתון שיוך לקבוצת מנהל. |
| setLicenseManager | משייכת משתמש נתון לקבוצת מנהל רשיונות |
| removeLicenseManager | מורידה משתמש נתון מקבוצת מנהל רשיונות |

TeamRequestsLocal

|  |  |
| --- | --- |
| מתודה | תאור |
| switchTeamManager | מקבלת כפרמטרים ID של צוות ושם משתמש, ומגדירה את המשתמש כראש הצוות. |
| createNewTeam | יוצרת צוות חדש במערכת |
| removeTeam | מוחקת צוות ממערכת |

## שכבת ממשק המשתמש – com.licensebox.ui

השכבה אחראית על הצגת ממשק המשתמש. הממשק של מערכת LicenseBox מורכב מדפי xhtml שכתובים בטכנולוגיית JSF Facelets. לכל דף (view) יש מחלקה (controller) שאחראית להביא ולעדכן את המידע משכבת ה- DB(module). מחלקות אלה הינן ManagedBeans של JSF ב-scope-ים שונים כגון SessionScoped לשמירת המידע לאורך ה-session של המשתמש ו-ViewScoped להצגת המידע ב-view המתאים. המחלקות נמצאות תחת חבילה **com.licensebox.ui.page**.

ממשק המשתמש מנסה לרתום את הכוח של AJAX בכדי לייעל ולשפר את חווית הלקוח. כתוצאה, זמן התגובה מתקצר והחוויה שנוצרת דומה לשימוש ב-desktop application רגיל.

הדפים בנויים על גבי Template בשם defaultTemplate.xhtml אשר מורכב משלושה חלקים. שניים מהם סטטיים: סרגל הניווט (nav.xhtml) וכותרת החוזרת בתחתית כל עמוד (footer.xhtml). בנוסף, ישנו החלק האמצעי אשר מתעדכן בכל עמוד. בצורה זו, הקוד יעיל יותר (אין צורך לחזור על אותו קוד מספר פעמים) ועובד בצורה חלקה יותר.

להלן הפירוט של ממשק המשתמש:

### התחברות למערכת

|  |  |  |
| --- | --- | --- |
| שם הדף | תאור | Controller |
| login.xhtml | הדף מציג מסך התחברות למערכת שמצפה לקבל יוזר וסיסמא של משתמש. |  |
| loginerror.xhtml | הדף מציג הודעת שגיאה למשתמש שמזין פרטי התחברות שגויים. |  |
| 403.xhtml | הדף מציג הודעת שגיאה למתשמש שאין לו הרשאה למערכת. |  |

### ממשק משתמש כללי

|  |  |  |
| --- | --- | --- |
| שם הדף | תאור | Controller |
| Index.xhtml | מסך ראשי של המערכת אשר מבצע את הפעולות הבאות: מציג את שם המשתמש המחובר ולאיזה צוות הוא שייך, משתמש גם רואה איזה רשיונות יש לו ואת הפאשרות להחזיר אותו. בנוסף, המשתמש מקבל רשימת בקשות לרשיונות חדשים הוא הגיש ומה הסטטוס שלהן. | DashboardManagedBean.java |
| Nav.xhtml | תפריט דינמי שמציג לכל משתמש את האפשרויות הניתנות לו לפי קבוצת הרשאה שאליה הוא שייך. ישנם ארבעה תפריטים אפשריים:   * תפריט לראש צוות ומציג רשיונות לאישור ודוחות, * תפריט למנהל שמציג רכישות לאישור ודוחות, * תפריט למנהל הרשיונות שמציג ניהול תוכניות, ניהול רשיונות, בקשות לאישור, בקשות לרכישה, דוחות ותפריט * תפריט למנהלן שנותן אפשרות לנהל יוזרים, צוותים ומציג לוג של כל הפעולות של כל המשתמשים במערכת |  |

### ממשק משתמש למנהלן

|  |  |  |
| --- | --- | --- |
| שם הדף | תאור | Controller |
| logsAdmin.xhtml | הדף מציג לוג של פעילות במערכת כגון התחברות של משתמש, יצירת בקשה חדשה או אישור. | LogsManagedBean.java |
| teamAdmin.xhtml | הדף מציג למנהלן את כל הצוותים המוגדרים במערכת ונותן אפשרות ליצור צוות חדש, לבצע עדכון או מחיקה של צוותים הקיימים. | TeamAdminManagedBean.java |
| userAdmin.xhtml | הדף מציג טבלה של כל המשתמשים הקיימים במערכת ואת פרטיהם. המנהלן יכול להוסיף משתמש, למחוק משתמש קיים, להוסיף משתמש קיים לקבוצת הרשאה מסוימת או לעדכן את פרטיו. | UserAdminManagedBean.java |

### ממשק משתמש למנהל הרשיונות

|  |  |  |
| --- | --- | --- |
| שם הדף | תאור | Controller |
| licenseManage.xhtml | הדף מציג רשימת כל הרישונות שנמצאות ברשותה של החברה. לכל רשיון מופיע לאיזה תוכנה הוא שייך, האם הוקצה למשתמש מסוים, תאריך הרכישה שלו ומחיר. כמו כן ניתנת אפשרות לצרף לרשיון קובץ PDF. המטרה היא לשמור לרשיון את המסמכים הנוגעים לרכישתו. | LicenseManagedBean.java  DownloadManagedBean.java |
| licenseManagerReports.xhtml | דוחות למנהל רשיונות. | LicenseManagerReportsManagedBean.java |
| programManage.xhtml | הדף מציג את כל התוכנות שנמצאות ברשות החברה. לכל תוכנה  מופיע שם, ID, גירסה ואתר. נין להוסיף תוכנה חדשה. | ProgramsManagedBean.java |
| purchaseRequests.xhtml | הדף מציג למנהל הרשיונות את הבקשות לרכש רשיונות חדשים שהוגשו למנהל ומה הסטטוס שלהן (האם אושרו). | PurchaseRequestsManagedBean.java |
| Requests.xhtml | הדף מציג את הבקשות להקצאת רשיון חדש למשתמש שמחכות לאישורו של מנהל הרשיונות. ניתן לאשר את הבקשה הרצויה. | LicenseRequestsManagedBean.java |

### תפריט למנהל

|  |  |  |
| --- | --- | --- |
| שם הדף | תאור | Controller |
| managerReports.xhtml | דוחות למנהל | ManagerReportsManagedBean.java |
| purchaseRequests.xhtml | הדף מציג את הבקשות לרכישת רשיון חדש שמחכות לאישורו של המנהל. ניתן לאשר את הבקשה הרצויה. | PurchaseApprovalManagedBean.java |

### תפריט לראש צוות

|  |  |  |
| --- | --- | --- |
| שם הדף | תאור | Controller |
| teamleadReports.xhtml | דוחות לראש הצוות | TeamLeaderReportsManagedBean.java |
| requests.xhtml | הדף מציג את הבקשות להקצאת רשיון חדש למשתמש שמחכות לאישורו של ראש הצוות. ניתן לאשר את הבקשה הרצויה. | TeamLeadLicenseApproveManagedBean.java |

מחלקות נוספות ששימשו לכתיבת ממשק משתמש:

## חבילה com.licensebox.ui.session

UserSessionManagedBean – מחלקה זו משמשת לניהול נתונים לאורך כל ה-session של המשתמש (עד שהוא מבצע Logout).

|  |  |
| --- | --- |
| מתודה | תאור |
| getUsername | מחזירה את שם המשתמש שאליו שייך ה-session |
| getFullname | מחזירה שם פרטי ושם משפחה של המשתמש |
| isAdministrator | מחזירה true אם המשתמש שייך לקבוצת מנהלן |
| isLicenseAdmin | מחזירה true אם המשתמש שייך לקבוצת מנהל רשיונות |
| isManager | מחזירה true אם המשתמש שייך לקבוצת מנהל |
| isTeamLeader | מחזירה true אם המשתמש שייך לקבוצת ראש צוות |

## חבילה com.licensebox.ui

Helper – מחלקה זו מכילה מספר מתודות סטטיות שנמצאות בשימוש ב-Managed Beans שפורטו לעיל

|  |  |
| --- | --- |
| מתודה | תאור |
| addFacesMessage | מוסיפה הודעת error או information שתוצג למשתמש. ההודעה מתווספת ל-context של JSF ומורכבת משני פרמטרים:   * חומרת ההודעה – INFO/ERROR (ישנם שני קבועים נומריים שמוגדרים במחלקה לצורך זה) * תוכן ההודעה |
| sortMapByKey | המתודה משמשת למיון טבלאות שמציגות רשימת תוכניות, רשיונות וכ"ו. |
| dateToCalendar | המתודה מבצעת המרה של אובייקט מסוג Date לאובייקט מסוג Calendar. |

## חבילה com.licensebox.ui.converter

החבילה מכילה מספר מחלקות שאחראיות להציג אובייקט כ-String והפוך. הם נמצאים בשימוש ב-Facelets (בקבצי xhtml) כאשר נדרשת המרה בין אובייקט לבין ייצוג הטקסטואלי שלו – הגדרה ב-attribute בשם converter בתגים תומכים.

המחלקות ממשות את הממשק Converter ואת הממשק Serializable. כל מחלקה מכילה שתי מתודות בלבד: getAsString ו-getAsObject.

להלן רשימת המחלקות בחבילה:

* AppRoleConverter – ממירה אובייקט של AppRole לייצוג ה-String וההיפך
* AppUserConverter – ממירה אובייקט של AppUser ומציגה את ערך ה-username
* PriceConverter – ממירה את ערך ה-price מ-String ל-BigDecimal וזורקת שגיאה במידה וההמרה לא הצליחה
* ProgramConverter – ממירה בין שדה שם התוכנה והגרסה שלה לאובייקט מסוג Program וההיפך
* ProgramIdConverter – ממירה את ערך ה-programId מ-String ל-Integer וזורקת שגיאה במידה וההמרה לא הצליחה
* TeamConverter – ממירה אובייקט של Team לייצוג ה-String וההיפך
* TeamIdConverter – ממירה את ערך ה-teamId מ-String ל-Integer וזורקת שגיאה במידה וההמרה לא הצליחה