|  |
| --- |
| האוניברסיטה הפתוחה |
| ShoeStore  מסמך תכנון |
| קורס: 20503 סדנה בתכנות מתקדם בשפת Java‏ |

|  |
| --- |
| שחר בר משה  מיכאל בולגר |

תוכן העניינים

[כללי 3](#_Toc461472981)

[טכנולוגיות 4](#_Toc461472982)

[תיאור טכנולוגיות 4](#_Toc461472983)

[בסיס הנתונים 5](#_Toc461472984)

[בסיס הנתונים 5](#_Toc461472985)

[ERD........................ 5](#_Toc461472986)

[תיאור מחלקות 6](#_Toc461472987)

[החבילה com.openu.model 6](#_Toc461472988)

[החבילה com.openu.controller 7](#_Toc461472989)

[controllers לאיסוף מידע על משתמש 9](#_Toc461472990)

[controllers המראים למשמש מידע מבסיס הנתונים 10](#_Toc461472991)

[החבילה com.openu.repository 12](#_Toc461472992)

[החבילה com.openu.security 12](#_Toc461472993)

[החבילה com.openu.service 13](#_Toc461472994)

[החבילה com.openu.util 14](#_Toc461472995)

[החבילה com.openu.validators 14](#_Toc461472996)

[תוכן WEB 15](#_Toc461472997)

[הגדרות 16](#_Toc461472998)

[קובץ תכונות 16](#_Toc461472999)

[הגדרות spring 16](#_Toc461473000)

[איתחול בסיס הנתונים 16](#_Toc461473001)

[הוראות התקנה והרצה 17](#_Toc461473002)

# כללי

מערכת זו מיועדת לניהול מכירות בחנות נעליים.

המערכת מספקת מענה לחנויות נעליים אשר רוצות למכור באמצעות אתר אינטרנט.

מסמך זה הוא סקירה טכנית של הפרויקט.

# טכנולוגיות

## תיאור טכנולוגיות

להלן תיאור הטכנולוגיות העיקריות שבעזרתם בוצע הפרויקט

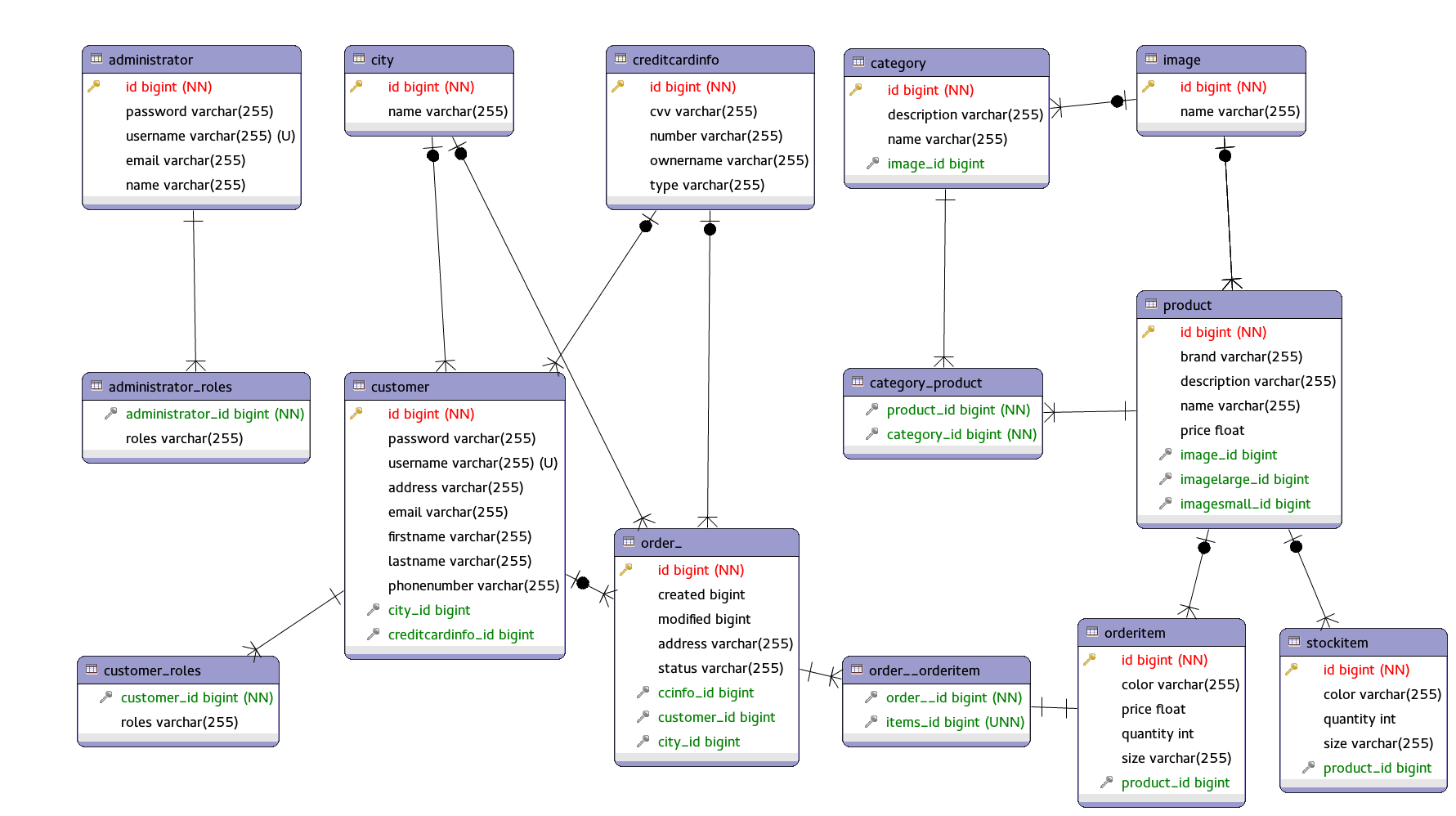
|  |  |
| --- | --- |
| שם חבילה | תיאור |
| Java 8 | שפת פיתוח |
| Apache MyFaces 2.2 | סביבת העבודה לפיתוח ממשק למשתמש |
| Tomahawk | ספריית עזר עבור רכיבי JSF שאינם כלולים ב-MyFaces |
| JSTL | ספריית טגיות עזר לבניית דפי JSF |
| Spring framework 4.2.5 | סביבת העבודה ופלטפורמה לניהול משאבי יישום.  נעשה שימוש ברכיבים הבאים:   * Spring core – רכיב מרכזי של הפלטפורמה המאפשר ניהול משאבים של היישום * Spring security – בקרת אימות המשתמשים ושליטה על הרשאות גישה * Spring data JPA – סביבת העבודה המשלימה את מימוש ה-JPA. מאפשר ניהול טרנזאקציות. * Spring AOP – מודול לביצוע פעולות aspect oriented * Spring Transactions – מודול לתמיכה בטרנזאקציות מול בסיסי הנתונים |
| C3P0 | ספרייה לניהול מאגרי חיבורים לבסיס הנתונים |
| Hibernate 5.0.5 | סביבת העבודה למיפוי יחסים בין אוביקטים הממשת את הטכנולוגיה JPA. |
| Maven | פלטפורמה לניהול תלויות ובניית הפרויקט |
| Apache Tomcat 8 | שרת היישום |
| PostgreSQL | בסיס הנתונים |
| Log4j 2 | ספריה אחראית על הדפסת לוגים של המערכת |
| Github | פלטפורמה לניהול גרסאות הקוד |

# בסיס הנתונים

## בסיס הנתונים

המערכת משתמשת בבסיס הנתונים PostgreSQL. המערכת מתקשרת עם בסיס הנתונים באמצעות טכנולוגית Hibernate (מימוש של JPA). יצירת הטבלאות מתבצעת אוטומטית מתוך ישויות המודל המוגדרים במערכת באמצעות מיפוי הישויות לטבלאות בעזרת אנוטציות ובעזרת ספריית HIBERNATE.

### ERD

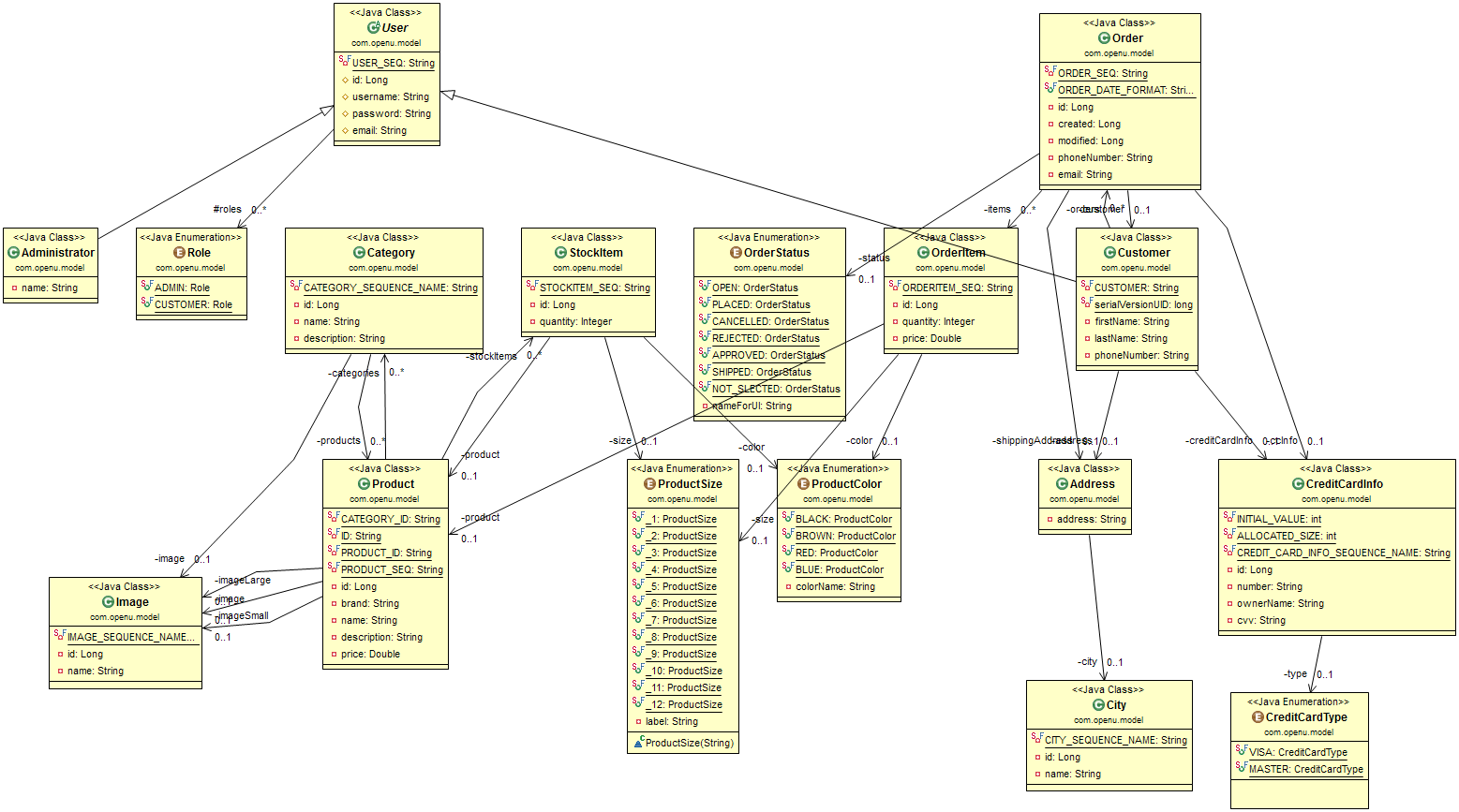


# תיאור מחלקות

## החבילה com.openu.model

תחת חבילה זו נמצאות מחלקות שמתארות את מודל הישויות

|  |  |
| --- | --- |
| שם מחלקה | תיאור |
| Address | המחלקה מייצגת ישות כתובת |
| Administrator | המחלקה מייצגת ישות משתמש מנהל |
| Category | המחלקה מייצגת ישות קטגוריית מוצרים |
| City | המחלקה מייצגת ישות עיר |
| CreditCardInfo | המחלקה מייצגת ישות נתוני כרטיס אשראי |
| CreditCardType | המחלקה מייצגת סוגים נתמכים של כרטיסי אשראי |
| Customer | המחלקה מייצגת ישות משתמש לקוח |
| Image | המחלקה מייצגת ישות תמונה |
| Order | המחלקה מייצגת ישות הזמנה |
| OrderItem | המחלקה מייצגת ישות פריט הזמנה |
| OrderStatus | המחלקה מייצגת סוגי סטטוסים של הזמנה |
| Product | המחלקה מייצגת ישות מוצר |
| ProductColor | המחלקה מייצגת סוגי צבע של מוצרים |
| ProductSize | המחלקה מייצגת סוגי מידה של מוצרים |
| Role | המחלקה מייצגת ישות תפקיד של משתמש |
| StockItem | המחלקה מייצגת ישות פריט מוצר במלאי |
| User | המחלקה מייצגת ישות משתמש |



## החבילה com.openu.controller

חבילה זו מכילה מחלקות controllers שמנהלות את הנתונים שיוצגו בדפי HTML

בתרשימים הבאים מופיע עץ המחלקות. בירוק- דפי HTML שהמחלקות מנהלות

|  |  |
| --- | --- |
| שם מחלקה | תיאור |
| ContactController | מחלקה זו מנהלת את התהליך "יצירת קשר" |
| CustomerController | מחלקה זו מנהלת פעולות הקשורות ללקוחות |
| CategoryController | מחלקה זו מנהלת פעולות שמבוצעות על ישויות קטגוריה |
| CheckoutController | מחלקה זו מנהלת את תהליך רכישת ההמנה |
| OrderController | מחלקה זו מנהלת פעולות הקשורות להזמנות |
| SessionBean | מחלקה זו מספקת מידע אודות לקוחות אקטיביים כרגע. במחלקה זו משתמשים גם controlles אחרים כservice |
| ProductController | מחלקה זו מנהלת פעולות הקשורות לישויות מוצרים |

### controllers לאיסוף מידע על משתמש

|  |  |
| --- | --- |
| שם מחלקה | תיאור |
| AbstractCustomerInformationCollector | מחלקת אב למחלקות controller אשר מנהלות עדכון פרטי לקוחות |
| AbstractInformationCollector | מחלקת אב למחלקות controller אשר מנהלות עדכון פרטי משתמשים |
| AdminEditInformationController | מחלקה זו מנהלת פעולות עדכון פרטי משתמשים מנהלים |
| EditInformationController | מחלקה זו מנהלת פעולות עדכון פרטי משתמשים |
| SignupController | מחלקה זו מנהלת תהליך הרישום לאתר |

### controllers המראים למשמש מידע מבסיס הנתונים

|  |  |  |
| --- | --- | --- |
| שם מחלקה | תיאור | פונקציות עיקריות |
| AbstractCrudController | מחלקת אב למחלקות controller שמיועדות לבצע פעולות קריאה, כתיבה, עדכון, מחיקה של ישות מסויימת. | * getAll() – פונקציה זו מחזירה את המידע המוצג בטבלאות של הcontrollers היורשים ממחלקה זו. הפונקציה מפעילה מיון או פילטור על התוצאה אם יש צורך |
| AbstractSearchController | מחלקת אב למחלקות controller אשר מבצעות פעולות חיפוש במוצרי החברה | CreatePredicateList() – פונקציה זו יוצרת את הפרדיקטים שלפיהם יבוצע החיפוש במוצרים |
| AdminCategoryController | מחלקה זו מנהלת פעולות יצירה, קריאה, עדכון, מחיקה של ישויות קטגוריה |  |
| AdminController | מחלקה זו מנהלת פעולות יצירה, קריאה, עדכון, מחיקה של ישויות משתמש מנהל |  |
| AdminCustomerController | מחלקה זו מנהלת פעולות יצירה, קריאה, עדכון, מחיקה של ישויות משתמש לקוח |  |
| AdminOrderController | מחלקה זו מנהלת פעולות שמבוצעות על ישויות הזמנה ע"י מנהל | * createPredicateList()- פונקציה זו יוצרת את הפרדיקטים בשלפיהם יבוצע הפילטור בטופס * cancelOrder,approveOrder ,rejectOrder, placeOrder - פעולות על הזמנות |
| AdminProductController | מחלקה זו מנהלת פעולות שמבוצעות על ישויות מוצרים ע"י מנהל |  |
| CityController | מחלקה זו מנהלת פעולות יצירה, קריאה, עדכון, מחיקה של ישויות עיר |  |
| SearchController | מחלקה זו מנהלת פעולות חיפוש במוצרים באתר הלקוח |  |

## החבילה com.openu.repository

חבילה זו מכילה מחלקות/ממשקים שמנהלים גישה לבסיס הנתונים.

|  |  |
| --- | --- |
| שם המחלקה | תיאור |
| AdministratorRepository | ממשק זה מגדיר פעולות לגישה לבסיס הנתונים עבור ישות משתמש המנהל |
| CategoryRepository | ממשק זה מגדיר פעולות לגישה לבסיס הנתונים עבור ישות קטגוריה |
| CityRepository | ממשק זה מגדיר פעולות לגישה לבסיס הנתונים עבור ישות עיר |
| CustomerRepository | ממשק זה מגדיר פעולות לגישה לבסיס הנתונים עבור ישות לקוח |
| ImageRepository | ממשק זה מגדיר פעולות לגישה לבסיס הנתונים עבור ישות תמונה |
| OrderRepository | ממשק זה מגדיר פעולות לגישה לבסיס הנתונים עבור ישות הזמנה |
| ProductRepository | ממשק זה מגדיר פעולות לגישה לבסיס הנתונים עבור ישות מוצר |
| StockItemRepository | ממשק זה מגדיר פעולות לגישה לבסיס הנתונים עבור ישות פריט מוצר במלאי |

## החבילה com.openu.security

חבילה זו מכילה מחלקות אשר מנהלות אוטנטיקציה ואוטוריזציה של משתמשים

|  |  |
| --- | --- |
| שם מחלקה | תיאור |
| AuthProvider | מחלקת זו נמצאת בשימוש ע"י סביבת העבודה של spring-security. מחלקה זו אחראית על ביצוע האוטנטיקציה של משתמשים |
| LoginManager | מחלקה מופעלת מדפי JSF ואחראית על ביצוע login ו-logout |

## החבילה com.openu.service

חבילה זו מכילה מחלקות שירות

|  |  |
| --- | --- |
| שם מחלקה | תיאור |
| AbstractOrderEmailSender | מחלקה אבסטרקטית משמשת כמחלקת אב לכל המחלקות האחראיות על שליחת אימייל בעקבות שינוי סטטוס הזמנה |
| ApprovedOrderEmailSender | מחלקה זו אחראית על שליחת אימייל בעקבות אישור ההזמנה ע"י המנהל |
| CancelOrderEmailSender | מחלקה זו אחראית על שליחת אימייל בעקבות ביטול ההזמנה ע"י המנהל |
| ContactEmailSender | מחלקה זו אחראית על שליחת אימייל בעקבות יצירת קשר ע"י משתמש |
| EmailSender | מחלקת אב למחלקות המנהולת שליחת אימייל |
| RejectedOrderEmailSender | מחלקה זו אחראית על שליחת אימייל בעקבות דחיית ההזמנה ע"י המנהל |
| ShippedOrderEmailSender | מחלקה זו אחראית על שליחת אימייל בעקבות אישור שליחת ההזמנה ע"י המנהל |
| createAdminUserEmailSender | מחלקה זו אחראית על שליחת אימייל למנהל חדש , עם פרטי ההתחברות למערכת |
| UserService | מחלקה זו אחראית על אתחול המשתמשים במערכת |

## החבילה com.openu.util

חבילה זו מכילה מחלקות עזר

|  |  |  |
| --- | --- | --- |
| שם מחלקה | תיאור | פונקציות עיקריות |
| AppContextProvider | מחלקה זו מאפשרת גישה למשאבים של spring ע"י מחלקות שאינן מנוהלות ע"י spring | * getApplicationContext() * setApplicationContext() |
| Constants | מחלקה זו מגדירה את המשתנים הקבועים ביישום |  |
| CustomerTransaction | מחלקה זו מגדירה אנוטציה שמאפשרת לסמן פעולות של משתמש שאמורות להתבצע בצורה סינכרונית |  |
| CustomerTransactionAspect | מחלקה זו ממשת את מנגנון הסנכרוניות שמופעל על מתודות המסומנות עם @CustomerTransaction |  |
| FilterManager | מחלקה זו מאפשרת למחלקות אחרות להגדיר ולבצע שאילתות דינאמיות | * getQueryResultList()- פונקציה זו מחזירה את תוצאת השאילתה , הכוללת בתוכה את כל הפרדיקטים שהוכנסו לצורך ביצוע הפילטר * פונקציות ליצירת פרדיקטים |
| NamedLock | מימוש של מנעול שמאפשר נעילה ופתיחה לפי שם כלשהו |  |
| Utils | מחלקה זו מכילה מתודות עזר כלליות | * getRequest()- פונקציה זו מחזירה את פרטי הrequest שנשלח מדף הHTML |
| ViewScope | מחלקה זו מגדירה תחום מחזור חיים של Spring Bean המותאם לאפליקציה. תחום זה מתנהג בדומה לתחום View של JSF. |  |

## החבילה com.openu.validators

חבילה זו מכילה מחלקות ולידציה של טפסי JSF

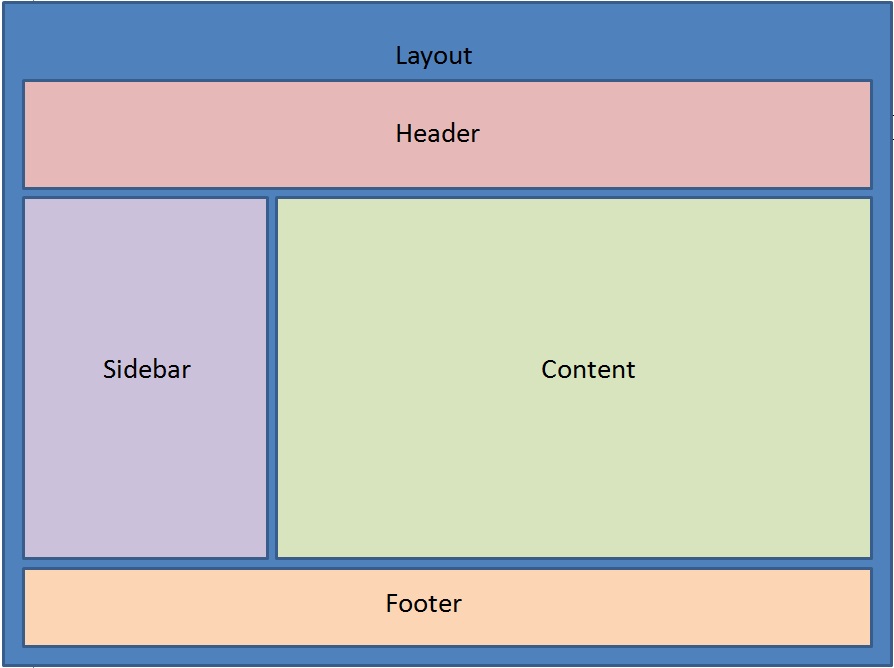
|  |  |
| --- | --- |
| שם מחלקה | תיאור |
| AcceptTermsValidator | מחלקה זו מבצעת ולידציה של שדה "accept terms" |
| CustomerUserNameValidator | מחלקה זו מבצעת ולידצית יחודיות שם משתמש |
| EmailValidator | מחלקה זו מבצעת ולידציה של שדות אימייל |
| PasswordAgainValidator | מחלקה זו מוודאת חוקיות של סיסמאות |
| PhoneValidator | מחלקה זו מוודאת תקינותם של שדות מספר טלפון |
| ProductAvailabilityValidator | מחלקה זו מוודאת המצאות של מוצר במלאי |

# תוכן WEB

פיתוח ממשק למשתמש בוצע בעזרת טכנולוגיות JSF (MyFaces).

דפי JSF נבנו בעזרת שימוש בבנית תבניות Facelets.

תבניות הדפים ממוקים תחת הספריה *template/src/main/webapp/template*, שאר דפי היישום מתבססים על התבניות הקיימות, כאשר באחראיותו של כל דף להגדיר את חלק התוכן (content) של התבנית.



ישנה הפרדה בין דפי המערכת שמיועדים ללקוחות האתר לבין הדפים של אתר מנהל.

דפי אתר המנהל נמצאים תחת הספריה *template/src/main/webapp/admin*.

דפי אתר הלקוחות נמצאים תחת הספריה *template/src/main/webapp/*.

# הגדרות

## קובץ תכונות

חלק ממאפייני היישום ניתן להגדיר דרך קובץ תכונות שנמצא תחת הנתיב *template/src/main/resources/props.properties*.

|  |  |
| --- | --- |
| שם מאפיין | תיאור |
| static.content.dir | ספריה מקומית על המחשב אשר מכילה תמונות שנמצאות בשימוש היישום |
| static.content.dir.web | שם ספריית התמונות שתהיה חשופה כלפי המשתמש |
| db.hibernate.dialect | ניב SQL אשר דרכו hibernate יתקשר עם בסיס הנתונים |
| db.show.sql | דגל שמורה האם להדפיס את שאילתות הSQL בקובצי לוג |
| db.url | שורת כתובת הגישה אל בסיס הנתונים |
| db.username | שם המשתמש של בסיס הנתונים |
| db.password | הסיסמא להתחברות לבסיס הנתונים |
| db.driver | מחלקת דרייבר שאיתו יתבצע החיבור לבסיס הנתונים |
| email.username | כתובת אימייל המנהלית של היישום, דרכה ישלחו האימיילים של המערכת |
| email.password | סיסמא של כתובת האימייל המנהלית |

## הגדרות spring

ספריות SPRING משמשות כסביבת העבודה המרכזית של היישום.

הקובץ *template/src/main/resources/application-context.xml* הוא הקובץ הראשי (entry point) של קונפיגורציית spring. בין היתר קובץ זה מגדיר כיצד SPRING מוצא משאבי מערכת, גישה לבסיס הנתונים, ניהול טרנזאקציות וכו'.

הקובץ *template/src/main/resources/security-application-context.xml* מכיל קונפיגורציית אימות והרשאות משתמשי המערכת.

## איתחול בסיס הנתונים

טבלאות בסיס הנתונים נוצרות ומתעדכנות אוטומטית בתוך המחלקות הנמצאות בחבילה *com.openu.model*. זה נעשה ע"י ספריית Hibernate.

ישנו איתחול ראשוני של היישום אשר דואג לאתחל טבלאות עם מידע נדרש, כגון רשימות ערים, קטגוריות, מוצרים. שאילתות DDL שמשתתפות באיתחול מוגדרות בקובץ *template/src/main/resources/script.sql.*

# הוראות התקנה והרצה

במסמך זה הוראות התקנה והרצה מותאמות בעיקר למערכות הפעלה Linux-like.

1. Install PostgreSQL DB
   1. yum install postgresql-server
   2. postgresql-setup initdb
   3. systemctl start postgresql
   4. su – postgres
   5. createuser -P openu (when prompted for password type "openu")
   6. createdb --owner=openu openu
2. Install Apache Tomcat 8
   1. Download <http://apache.mivzakim.net/tomcat/tomcat-8/v8.5.5/bin/apache-tomcat-8.5.5.zip>
   2. Unzip under {tomcat\_home}
3. Install Maven
   1. > cd {project\_directory}
   2. >mvn clean install
4. Create directory “static” as /home/{user}/static; it will contain all uploaded images
5. Copy {project\_directory}/src/main/webapp/static/\* into /home/{user}/static/
   1. Edit {tomcat\_home}/conf/server.xml
   2. Add the following in the file: <Host appBase="webapps"> <Context docBase="/home/{user}/static" path="/template/static" /> </Host>
6. Run the application
   1. Copy {project\_dir}/target/template.war into {tomcat\_home}/webapps/
   2. Execute the file {tomcat\_home}/bin/startup.sh
7. Browse to <http://localhost:8080/template>