

האוניברסיטה הפתוחה

ShoeStore

מסמך תכנון

קורס: 20503 סדנה בתכנות מתקדם בשפת Java

שחר בר משה

מיכאל בולגר

9/14/2016

תוכן העניינים

3 כללי
4 טכנולוגיות
4 תיאור טכנולוגיות
5 בסיס הנתונים
5 בסיס הנתונים
	5 ERD
6 תיאור מחלקות
6 החבילה com.openu.model
7 החבילה com.openu.controller
8 החבילה com.openu.repository
8 החבילה com.openu.security
9 החבילה com.openu.service
9 החבילה com.openu.util
9 החבילה com.openu.validators
11 תוכן WEB
12 הגדרות
12 קובץ תכונות
12 הגדרות spring
12 איתחול בסיס הנתונים
13 הוראות התקנה והרצה

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

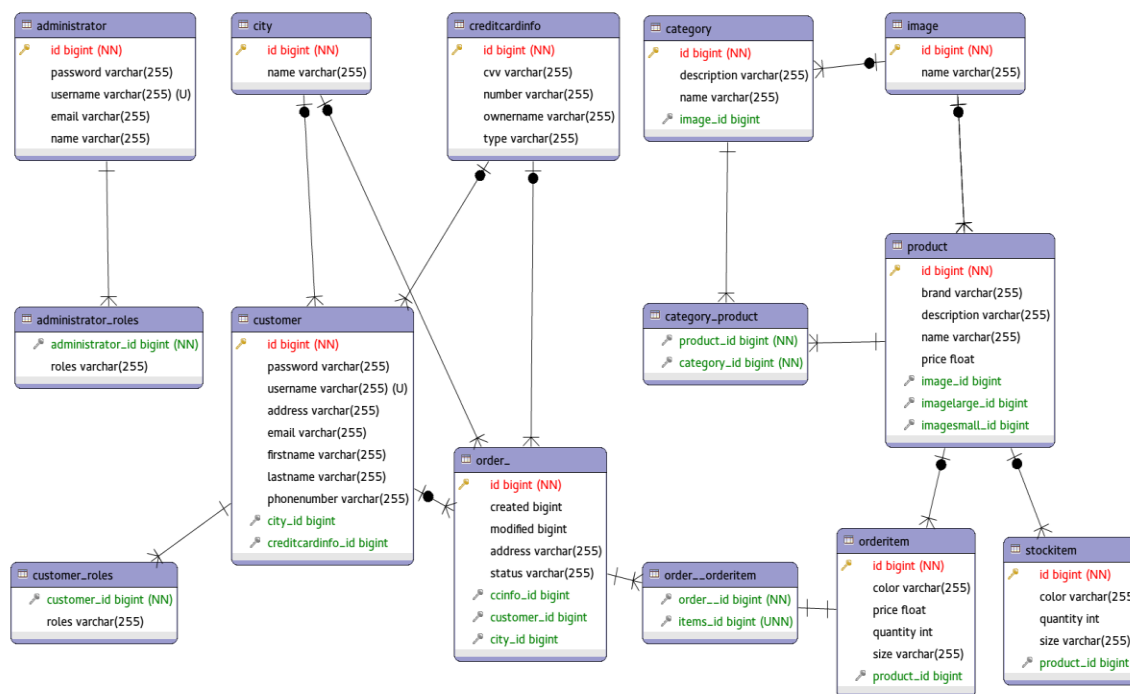
שם חבילה	תיאור
Java 8	שפת פיתוח
Apache MyFaces 2.2	סביבת העבודה לפיתוח ממשק למשתמש
Tomahawk	ספריית עזר עבור רכיבי JSF שאינם כלולים ב-MyFaces
JSTL	ספריית טגיות עזר לבניית דפי JSF
Spring framework 4.2.5	<p>סביבת העבודה ופלטפורמה לניהול משאבי יישום. נעשה שימוש ברכיבים הבאים:</p> <ul style="list-style-type: none"> • <u>Spring core</u> – רכיב מרכזי של הפלטפורמה המאפשר ניהול משאבים של היישום • <u>Spring security</u> – בקרת אימות המשתמשים ושליטה על הרשאות גישה • <u>Spring data JPA</u> – סביבת העבודה המשלימה את מימוש ה-JPA. מאפשר ניהול טרנזאקציות. • <u>Spring AOP</u> – מודול לביצוע פעולות aspect oriented • <u>Spring Transactions</u> – מודול לתמיכה בטרנזאקציות מול בסיסי הנתונים
C3P0	ספרייה לניהול מאגרי חיבורים לבסיס הנתונים
Hibernate 5.0.5	סביבת העבודה למיפוי יחסים בין אובייקטים הממשת את הטכנולוגיה JPA.
Maven	פלטפורמה לניהול תלויות ובניית הפרויקט
Apache Tomcat 8	שרת היישום
PostgreSQL	בסיס הנתונים
Log4j 2	ספרייה אחראית על הדפסת לוגים של המערכת
Github	פלטפורמה לניהול גרסאות הקוד

בסיס הנתונים

בסיס הנתונים

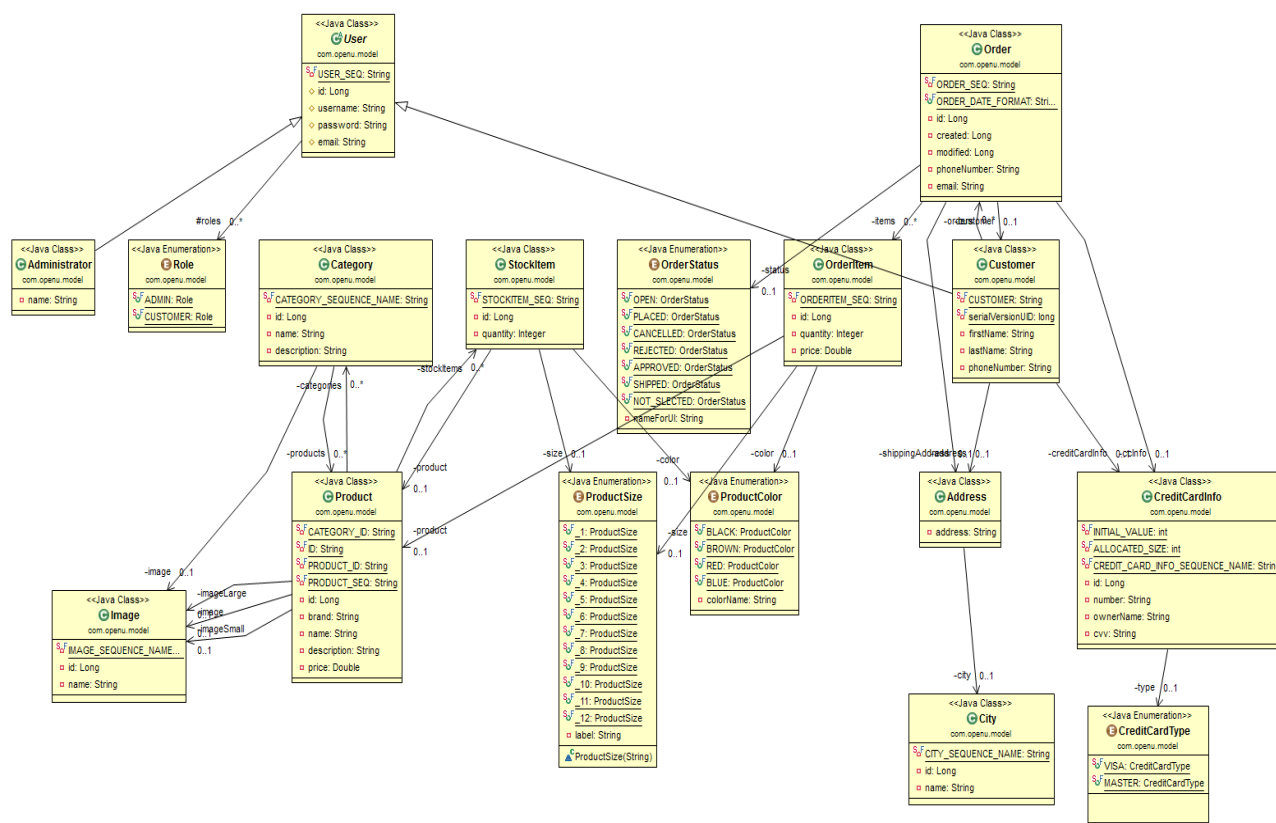
המערכת משתמשת בבסיס הנתונים PostgreSQL. המערכת מתקשרת עם בסיס הנתונים באמצעות טכנולוגיית Hibernate (מימוש של JPA). יצירת הטבלאות מתבצעת אוטומטית מתוך ישויות המודל המוגדרים במערכת באמצעות מיפוי הישויות לטבלאות בעזרת אנוטציות ובעזרת ספריית HIBERNATE.

ERD



תחת חבילה זו נמצאות מחלקות שמתארות את מודל הישויות

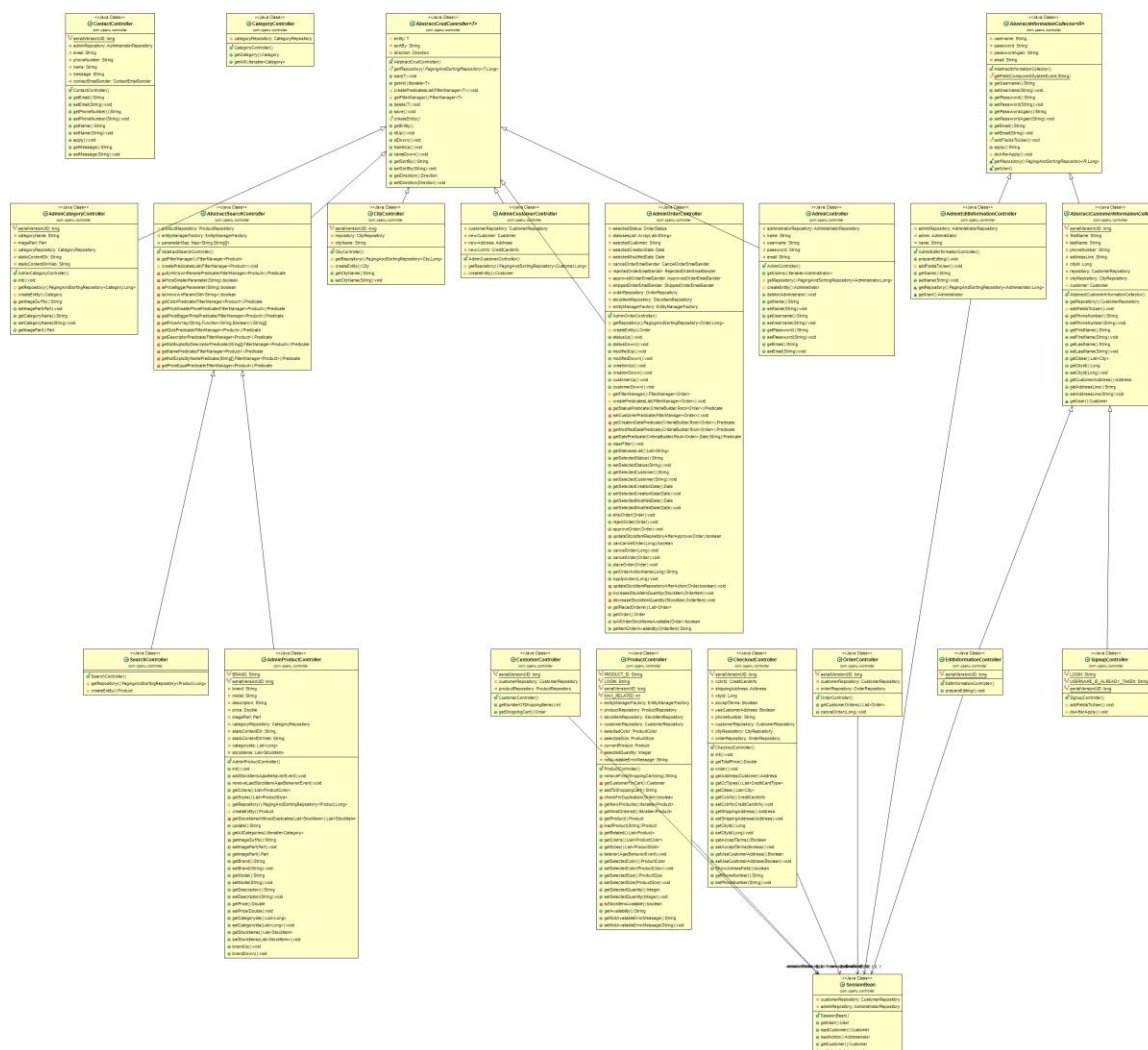
שם מחלקה	תיאור
Address	המחלקה מייצגת ישות כתובת
Administrator	המחלקה מייצגת ישות משתמש מנהל
Category	המחלקה מייצגת ישות קטגוריית מוצרים
City	המחלקה מייצגת ישות עיר
CreditCardInfo	המחלקה מייצגת ישות נתוני כרטיס אשראי
CreditCardType	המחלקה מייצגת סוגים נתמכים של כרטיסי אשראי
Customer	המחלקה מייצגת ישות 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

שם מחלקה	תיאור
AbstractCrudController	מחלקת אב למחלקות controller שמיועדות לבצע פעולות קריאה, כתיבה, עדכון, מחיקה של ישות מסוימת.
AbstractCustomerInformationCollector	מחלקת אב למחלקות controller אשר מנהלות עדכון פרטי לקוחות
AbstractInformationCollector	מחלקת אב למחלקות controller אשר מנהלות עדכון פרטי משתמשים
AbstractSearchController	מחלקת אב למחלקות controller אשר מבצעות פעולות חיפוש
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
AdminEditInformationController	מחלקה זו מנהלת פעולות עדכון פרטי משתמשים מנהלים
AdminOrderController	מחלקה זו מנהלת פעולות שמבוצעות על ישויות הזמנה ע"י מנהל
AdminProductController	מחלקה זו מנהלת פעולות שמבוצעות על ישויות מוצרים ע"י מנהל
CategoryController	מחלקה זו מנהלת פעולות שמבוצעות על ישויות קטגוריה
CheckoutController	מחלקה זו מנהלת את תהליך רכישת ההמנה
CityController	מחלקה זו מנהלת פעולות יצירה, קריאה, עדכון, מחיקה של ישויות עיר
ContactController	מחלקה זו מנהלת את התהליך "צירת קשר"
CustomerController	מחלקה זו מנהלת פעולות הקשורות ללקוחות
EditInformationController	מחלקה זו מנהלת פעולות עדכון פרטי משתמשים
OrderController	מחלקה זו מנהלת פעולות הקשורות להזמנות
ProductController	מחלקה זו מנהלת פעולות הקשורות לישויות מוצרים
SearchController	מחלקה זו מנהלת פעולות חיפוש
SessionBean	מחלקה זו מספקת מידע אודות לקוחות אקטיביים כרגע
SignupController	מחלקה זו מנהלת תהליך הרישום לאתר



COM.OPENU.REPOSITORY החבילה

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

COM.OPENU.SECURITY החבילה

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 המנהלות 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
UserService	מחלקה זו אחראית על איתחול המשתמשים במערכת

החבילה COM.OPENU.UTIL

חבילה זו מכילה מחלקות עזר

שם מחלקה	תיאור
ApplicationContextProvider	מחלקה זו מאפשרת גישה למשאבים של spring ע"י מחלקות שאינן מנהלות ע"י spring
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 ממששת את מנגנון הסנכרוניות שמופעל על מתודות המסומנות עם @CustomerTransaction
FilterManager	מחלקה זו מאפשרת למחלקות אחרות להגדיר ולבצע שאילתות דינאמיות
NamedLock	מימוש של מנועל שמאפשר נעילה ופתיחה לפי שם כלשהו
Utils	מחלקה זו מכילה מתודות עזר כלליות
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

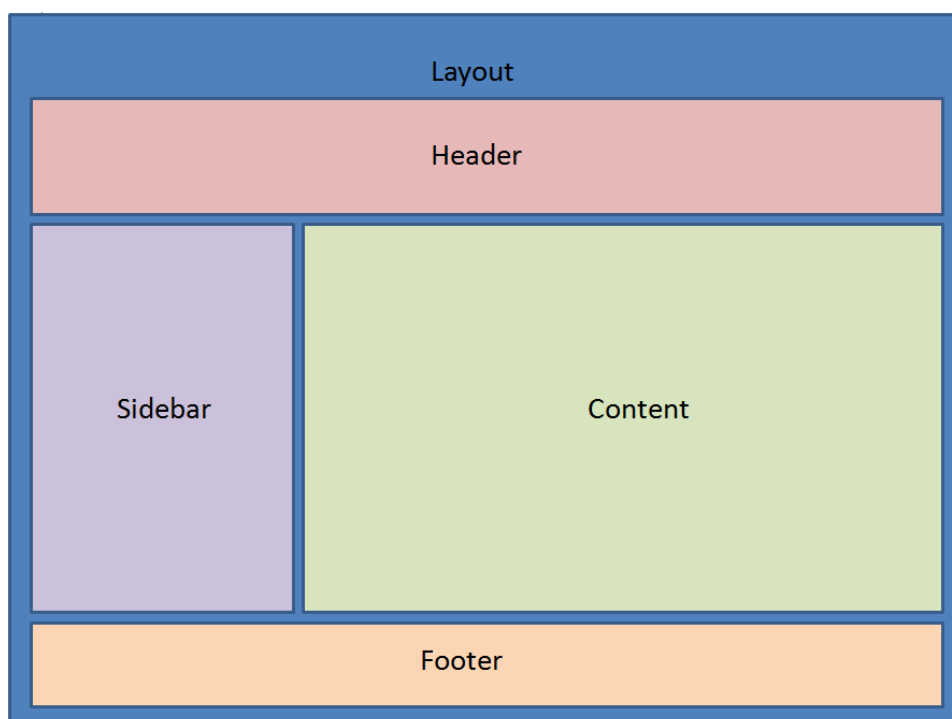
שם מחלקה	תיאור
AcceptTermsValidator	מחלקה זו מבצעת ולידציה של שדה "accept terms"

מחלקה זו מבצעת ולידצית יחודיות שם משתמש	CustomerUserNameValidator
מחלקה זו מבצעת ולידציה של שדות אימייל	EmailValidator
מחלקה זו מוודאת חוקיות של סיסמאות	PasswordAgainValidator
מחלקה זו מוודאת תקינותם של שדות מספר טלפון	PhoneValidator
מחלקה זו מוודאת המצאות של מוצר במלאי	ProductAvailabilityValidator

פיתוח ממשק למשתמש בוצע בעזרת טכנולוגיות JSF (MyFaces).

דפי JSF נבנו בעזרת שימוש בבניית תבניות Facelets.

תבניות הדפים ממוקמים תחת הספרייה `template/src/main/webapp/template`, שאר דפי היישום מתבססים על התבניות הקיימות, כאשר באחריותו 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 הספרייה `template/src/main/webapp/admin`.

דפי אתר הלקוחות נמצאים תחת הספרייה `template/src/main/webapp/`.

הגדרות

קובץ תכונות

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

שם מאפיין	תיאור
<code>static.content.dir</code>	ספריה מקומית על המחשב אשר מכילה תמונות שנמצאות בשימוש היישום
<code>static.content.dir.web</code>	שם ספריית התמונות שתהיה חשופה כלפי המשתמש
<code>db.hibernate.dialect</code>	ניב SQL אשר דרכו hibernate יתקשר עם בסיס הנתונים
<code>db.show.sql</code>	דגל שמורה האם להדפיס את שאילתות הSQL בקובצי לוג
<code>db.url</code>	שורת כתובת הגישה אל בסיס הנתונים
<code>db.username</code>	שם המשתמש של בסיס הנתונים
<code>db.password</code>	הסיסמא להתחברות לבסיס הנתונים
<code>db.driver</code>	מחלקת דרייבר שאיתו יתבצע החיבור לבסיס הנתונים
<code>email.username</code>	כתובת אימייל המנהלית של היישום, דרכה ישלחו האימיילים של המערכת
<code>email.password</code>	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
 - 1.1 yum install postgresql-server
 - 1.2 postgresql-setup initdb
 - 1.3 systemctl start postgresql
 - 1.4 su – postgres
 - 1.5 createuser -P openu (when prompted for password type "openu")
 - 1.6 createdb --owner=openu openu
2. Install Apache Tomcat 8
 - 2.1 Download <http://apache.mivzakim.net/tomcat/tomcat-8/v8.5.5/bin/apache-tomcat-8.5.5.zip>
 - 2.2 Unzip under {tomcat_home}
3. Install Maven
 - 3.1 > cd {project_directory}
 - 3.2 > 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/
 - 5.1 Edit {tomcat_home}/conf/server.xml
 - 5.2 Add the following in the file: <Host appBase="webapps"> <Context docBase="/home/{user}/static" path="/template/static" /> </Host>
6. Run the application
 - 6.1 Copy {project_dir}/target/template.war into {tomcat_home}/webapps/
 - 6.2 Execute the file {tomcat_home}/bin/startup.sh
7. Browse to <http://localhost:8080/template>