

# Adopt me



## מסמר תכנון

## **תוכן עניינים:**

3.....	<a href="#"><u>תיאור כללי</u></a>
3.....	<a href="#"><u>מינוחים בשימוש במסמך</u></a>
3.....	<a href="#"><u>טכנולוגיות בשימוש במערכת</u></a>
3.....	<a href="#"><u>נקודת כניסה למערכת</u></a>
3.....	<a href="#"><u>בסיס הנתונים</u></a>
4.....	<a href="#"><u>גישה לבסיס הנתונים</u></a>
4.....	<a href="#"><u>הטבלאות בבסיס הנתונים</u></a>
5.....	<a href="#"><u>ERD של המערכת</u></a>
6.....	<a href="#"><u>תרשים כללי של המערכת</u></a>
7.....	<a href="#"><u>ספריות JAVA עיקריות בשימוש</u></a>
8.....	<a href="#"><u>מחלקות ה-JAVA במערכת</u></a>
8.....	<a href="#"><u>תיאור קשרים עיקריים במערכת</u></a>
9.....	<a href="#"><u>חבילות במערכת</u></a>

## תיאור כללי

המסמך מתאר את מבנה המערכת שמרכיבה את אתר Adopt Me ואת הטכנולוגיות שבשימוש.

## מינוחים בשימוש במסמך

משתמש קצה- משתמש במערכת שהוא קהל היעד של המערכת ומפרסם חיות לאימוץ או מתעניין בתרומה, בהתנדבות או באימוץ.

## טכנולוגיות בשימוש במערכת

האתר משלב בתוכו מספר טכנולוגיות:

- MySQL Server - מכיל את טבלאות הנתונים של המערכת.
- JDBC - מאפשר גישה לבסיס הנתונים.
- Java classes, Servlets - קוד JAVA שרץ בצד השרת.
- קבצי ts - קבצי TypeScript שבנויים על JavaScript ומתממשים ביחד עם קבצי ה-HTML.
- Tomcat, JRE - שרת Web המריץ את ה-Servlets וקבצי ה-ts.
- Bootstrap - סביבת עבודה בקוד פתוח לצד לקוח, שמכילה אוסף של כלים ליצירת אפליקציות רשת ואתרים.
- JPA - ממשק העוזר בניהול בסיסי מידע רלציונליים באפליקציות הכתובות ב-JAVA.

## נקודת כניסה למערכת

הגישה למערכת מתבצעת באמצעות קריאת HTTP מול שרת ה-Angular שמאזין לפורט 4200 במחשב הלוקאלי. ניתן לגשת למערכת באמצעות <http://localhost:4200>. כאשר ניגשים לכתובת זאת, מורץ ה-app.component של האפליקציה שזו נקודת ההתחלה.

## בסיס הנתונים

בסיס הנתונים ממומש באמצעות MySQL Server שמאזין על TCP/IP כאשר פורט ברירת המחדל הוא 3306 וניתן לגשת אליו באמצעות האפליקציה הייעודית MySQL Workbench.

## גישה לבסיס הנתונים

הגישה לבסיס הנתונים מתבצעת על ידי הרצה של MySQL Server. יצירת ה-connection מתבצעת באמצעות JPA וביצוע השאילתות מתקיימת באמצעות שכבת האבסטרקציה של Hibernate.

## הטבלאות בבסיס הנתונים

מאגר הנתונים נקרא בשם adoptme והוא מכיל את הטבלאות הבאות:

שם הטבלה	הגדרת הטבלה
animal	הטבלה מכילה את כל החיות המוצעות לאימוץ באתר. כל חיה מכילה נתונים שונים אודותיה כגון שם, תיאור, סוג וכו'. מפתח: ID.
contact_request	הטבלה מכילה את כל בקשות יצירת הקשר שהוגשו על ידי משתמשי קצה באתר. כל בקשה מכילה את פרטי המשתמש כגון שם ואימייל. מפתח: ID.
donation	הטבלה מכילה את כל בקשות התרומה שהוגשו על ידי משתמשי קצה באתר. כל בקשה מכיל את פרטי התורם וכן את הסכום הרצוי לתרומה. מפתח: ID.
user	הטבלה מכילה את כל משתמשי האתר הרשומים. כל משתמש מכיל את פרטיו כגון שם פרטי ושם משפחה, כתובת וכו'. מפתח: ID.
volunteer	הטבלה מכילה את כל בקשות ההתנדבות שהוגשו על ידי משתמשי הקצה באתר. כל בקשה מכילה את פרטי המתנדב כגון שם ואימייל. מפתח: ID.

## ERD

animal	
PK	id
	animal_name image_url age description animal_kind contact_name contact_phone

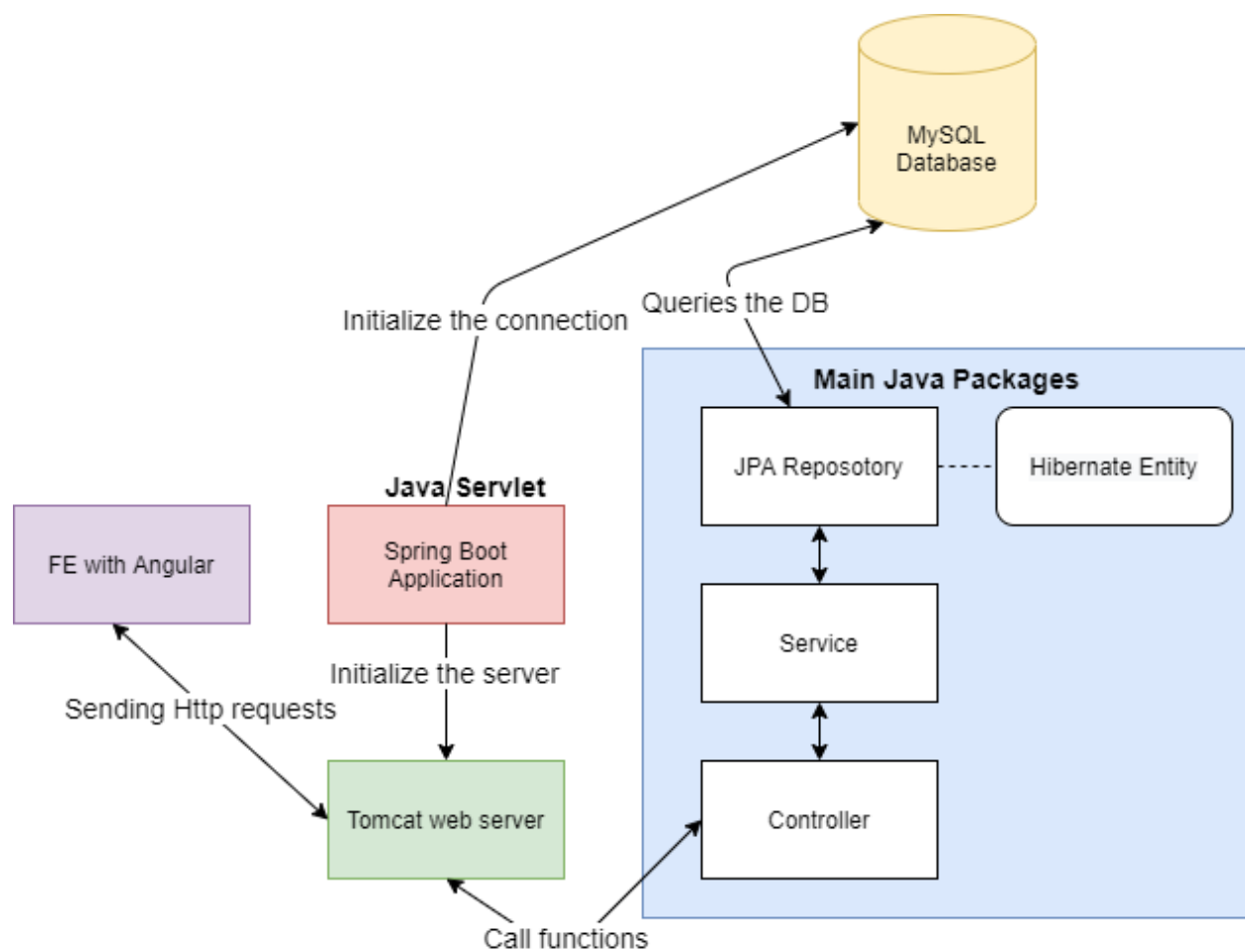
contact_request	
PK	id
	address city email first_name last_name phone

donation	
PK	id
	address city email first_name last_name phone amount donation_date

user	
PK	id
	address city email first_name last_name phone password

volunteer	
PK	id
	address city email first_name last_name phone

## תרשים כללי של המערכת

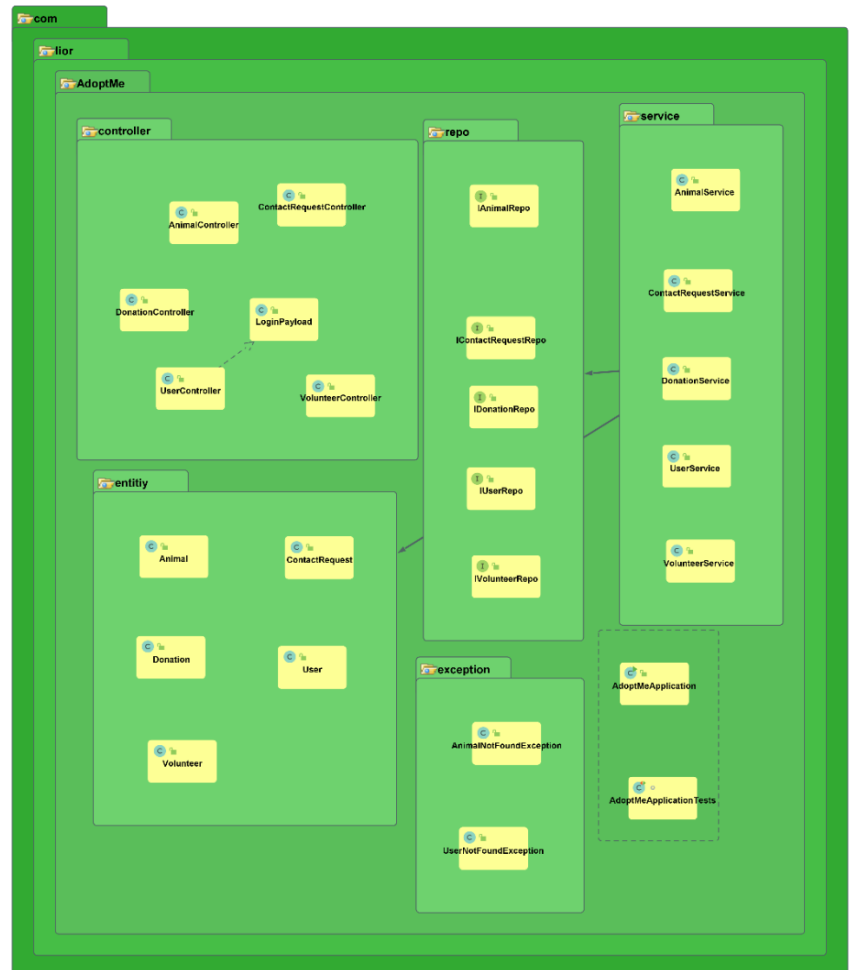
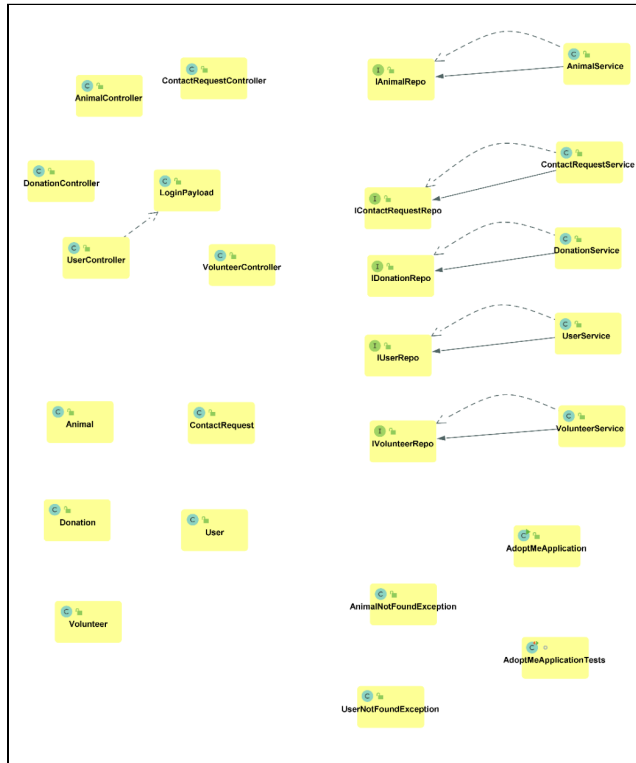


## ספריות JAVA עיקריות בשימוש

שם ספרייה	תאור
JpaRepository	ספרייה זו שייכת לספריית Spring Data. באמצעות ספרייה זו ניתן לבצע שאלות על בסיס מידע רלציוני (כדוגמת MySQL). הספרייה מבצעת מיפוי בין אובייקטים בקוד לאובייקטים בתוך בסיס הנתונים ובכך מאפשרת התממשקות קלה וכן ביצוע שאלות באמצעות קוד Java.
Spring Framework	באמצעות Spring ניתן לבצע פעולות רבות מהתממשקות לDB, העלאת שרתי Tomcat Web ועוד. בפרויקט השתמשתי בספרייה בכל ה"ל באמצעות Spring boot להעלאת השרתים, בתת ספרייה שקשורה ל-persistence לביצוע פעולות על DB וכן בתת הספרייה של HTTP לביצוע מיפוי עבור routing של קריאות API השונות.
javax.persistence	בספרייה זו השתמשתי על מנת לייצר אובייקטים המקיימים את עקרונות ה-OOP אשר מקושרים לאובייקטים בDB. באמצעות שימוש זה ניתן לבצע פעולות בפשטות על בסיס הנתונים באמצעות פקודות java רגילות.
MySQL Framework	ספרייה המאפשרת לבצע התחברות לבסיס הנתונים מבוסס MySQL.

## JAVA-מחלקות

## תיאור קשרים עיקריים במערכת





## חבילת ה-Controller

בחבילת ה-Controller נמצאים כל ה-controllers של הפרוייקט המבצעים את המיפוי בין קריאות ה-HTTP לבין הפונקציונליות במערכת. לכל דף יש routing שונה ועל כן controller שונה. להלן רשימת ה-Controllers:

- מחלקת AnimalController
- מחלקת ContactRequestController
- מחלקת DonationController
- מחלקת LoginPayload
- מחלקת UserController
- מחלקת VolunteerController

## חבילת ה-Entity

בחבילת ה-Entity נמצאים כל היישויות של הפרוייקט המתארות את הטבלאות הקיימות ב-DB. כל entity מכיל את כל השדות של היישות המתפקדות כעמודות באותה טבלה ב-DB. להלן רשימת היישויות:

- מחלקת Animal
- מחלקת ContactRequest
- מחלקת Donation
- מחלקת User
- מחלקת Volunteer

## חבילת ה-Repo

בחבילת ה-Repo נמצאים כל הממשקים המממשים את JpaRepository ובכך יוצרים ממשק עם ה-DB ומאפשרים לבצע שאילתות כל אותה יישות שהוא ממש. להלן רשימת הממשקים:

- מחלקת IAnimalRepo
- מחלקת IContactRequestRepo
- מחלקת IDonationRepo
- מחלקת IUserRepo
- מחלקת IVolunteerRepo

### חבילת ה-Service

בחבילת ה-service נמצאים כל השירותים המשמשים את ה-contollers המתאימים לבצע את הפונקציונליות הנדרשת באמצעות הפעלת פונקציות של JpaRepository. להלן רשימת ה-services:

- מחלקת AnimalService
- מחלקת ContactRequestService
- מחלקת DonationService
- מחלקת UserService
- מחלקת VolunteerService

### חבילת ה-Exception

בחבילת ה-exception נמצאות שתי מחלקות המתארות שגיאות הקשורות לאי מציאת entity מסויים ב-DB. להלן רשימת ה-exceptions:

- מחלקת AnimalNotFoundException
- מחלקת UserNotFoundException