



HOSPITAL APPOINTMENT SYSTEM

מסמך תכנון



APRIL 16, 2023

גיל אלפרט ומאור שווקי
סדנה בתכנות מתקדם בשפת ג'אווה



1 תוכן עניינים

1 תוכן עניינים	1
3 תיאור המערכת	2
3 טכנולוגיות בפרויקט	2.1
3 בסיס הנתונים	2.2
3 appointments טבלת	2.2.1
3 doctors טבלת	2.2.2
4 users טבלת	2.2.3
4 working_hours טבלת	2.2.4
5 Entity-Relations דיאגרמת	2.2.5
6 דיאגרמה מבנה כללי	2.3
7 (Class Diagram) דיאגרמת מחלקות	2.4
8 דפי האתר	3
8 admin דף	3.1
8 appointment-download דף	3.2
8 appointmentFailure דף	3.3
9 appointmentSuccess דף	3.4
9 index דף	3.5
9 login דף	3.6
9 make-appointment דף	3.7
10 myprofile דף	3.8
12 signup דף	3.9
13 Web תוספות	4
13 XML קבצי קונפיגורציה	4.1
13 JS קבצי	4.2
13 CSS קבצי	4.3
13 Assets קבצי	4.4
14 JSF's Managed Bean מחלקות	5
14 AdminBean מחלקת	5.1
14 AppointmentExportBean מחלקת	5.2
14 AppointmentFormBean מחלקת	5.3
15 AppointmentsCalendarBean מחלקת	5.4
16 DoctorListBean מחלקת	5.5
16 LoginBean מחלקת	5.6





16.....	ProfileAppointmentsBean	מחלקת	5.7
17.....	ProfileDataBean	מחלקת	5.8
17.....	ProfileDoctorBean	מחלקת	5.9
18.....	SignupBean	מחלקת	5.10
18.....	SuccessAppointmentBean	מחלקת	5.11
18.....	UserRedirectionBean	מחלקת	5.12
20.....	JPA's Entity	מחלקות	6
20.....	AppointmentsEntity	מחלקת	6.1
20.....	DoctorsEntity	מחלקת	6.2
20.....	UsersEntity	מחלקת	6.3
21.....	WorkingHoursEntity	מחלקת	6.4
21.....	WorkingHoursEntityPK	מחלקת	6.5
22.....	רכיבים נוספים		7
22.....	DBHandler	מחלקת	7.1
22.....	Pair	מחלקת	7.2
23.....	PDFHandler	מחלקת	7.3
23.....	AppointmentExporter	מחלקת	7.4
24.....	Utils	מחלקת	7.5
24.....	BirthDateValidator	מחלקת	7.6
24.....	DateConverter	מחלקת	7.7





2 תיאור המערכת

בפרק זה נתאר את מבנה הפרויקט מרמת ביניים. נראה את הטכנולוגיות המשומשות בפרויקט, נכיר את מבנה בסיס הנתונים, ונראה מספר תרשימים הממחישים את מבנה המערכת וזרימת המידע בה.

2.1 טכנולוגיות בפרויקט

מערכת קביעת התורים משתמשת בטכנולוגיות הבאות:

- JDK 18.0 – כלי הפיתוח הסטנדרטיים של Java.
- JSF 2.3 – Java Server Faces, הטכנולוגיה לשילוב ג'אווה לייצור דפי HTML רגילים.
- AJAX – טכנולוגיה לבניית דפי אינטרנט אינטראקטיביים, דינמיים, ומודרניים ע"י טעינת רכיבים ללא טעינת דף שלם מחדש.
- JavaScript – שפת JavaScript מסייעת לנו בניווטים ותכנות Web בסיסי.
- CSS – שפה מבוססת תכונות לעיצוב האתר.
- MySQL – שירות אירוח לבסיס הנתונים, גרסה 8.0.
- JDBC – מאפשר גישה לבסיס הנתונים על ידי קוד Java ו-Driver.
- JPA – Java Persistence API, עבודה נוחה עם טבלאות בסיס הנתונים בתור עצמי Entity.
- GlassFish 5.2022.5 – שרת המארח את האתר. מורץ באופן מקומי על מחשבנו האישי.

בהמשך המסמך מתוארים השימושים בטכנולוגיות הללו והקשרים ביניהן.

2.2 בסיס הנתונים

כמו שרת ה-GlassFish שמארח את האתר, גם השרת של MySQL Server מתארח אצל המחשב האישי שלנו ומארח את בסיס הנתונים. השרת מאזין על Port ברירת המחדל, 3306, בכתובת "jdbc:mysql://localhost: 3306/hospital". בסיס הנתונים מקבל חיבורים) משרת ה-GlassFish אשר אצלו מתארח האתר.

כעת, נתאר את טבלאות בסיס הנתונים "hospital" ואת הקשרים ביניהן בדיאגרמת ERD.

2.2.1 טבלת appointments

הטבלה שומרת את כל התורים/פגישות שנקבעו אי פעם במערכת, גם אלה שבוטלו (ואותן מסומנת בשדה המבוטלות). תחילה, בנינו טבלה זאת כך שהמפתח הראשי יהיה הזוג: (doctor_id, start_time), אך נתקלנו בבעיה ושמנו לב שמפתח ראשי זה אינו מספק את הפונקציונליות שרצינו: לא תיתכנה שתי פגישות מבוטלות של אותו רופא שהתחילו באותו הזמן. לכן זנחנו רעיון זה ועברנו לזיהוי הפגישות בעזרת מזהה id ייחודי.

שם השדה	חתימת השדה
id	int not null unique primary key
start_time	timestamp not null
end_time	timestamp not null
doctor_id	varchar(9) not null
patient_id	varchar(9) not null
description	varchar(1000)
is_cancelled	bit not null default 0

2.2.2 טבלת doctors

כל הרופאים מיוצגים בתור משתמשים רגילים, ולכן טבלה זו נותנת מידע נוסף (ולא במקום) על הרופאים. כמו כן, ה-id בטבלת הרופאים, שהינו המפתח הראשי, מהווה מפתח זר לטבלת המשתמשים שנראה בהמשך, ולכן עצם קיום טבלה זו קובע מי מהמשתמשים הינו רופא ומי לא.



שם השדה	חתימת השדה
id	int not null unique primary key
type	varchar(20) not null
about	varchar(200)
city	varchar(20) not null
-	foreign key (id) references Users(id) on delete cascade

כמובן שלצרכים מעשיים יותר בעולם האמיתי, ניתן להתאים טבלה זו בקלות כך שתכיל נתונים משמעותיים יותר. למשל, להפוך את שדה ה-city למיקוד/קואורדינטות ולהיעזר ב-API של מפת העולם על מנת לקבל מכך את המידע הדרוש.

2.2.3 טבלת users

טבלת users מכילה מידע פשוט יחסית על כל המשתמשים במערכת. רופאים ומנהלים גם נחשבים בתור משתמש ולכן גם חייבים להופיע בשורה בטבלה זאת.

שם השדה	חתימת השדה
id	int not null unique primary key
first_name	varchar(20) not null
last_name	varchar(20) not null
birth_date	date not null
password	varchar(16) not null
phone_number	varchar(10)
email	varchar(320)
is_admin	bit not null default 0

2.2.4 טבלת working_hours

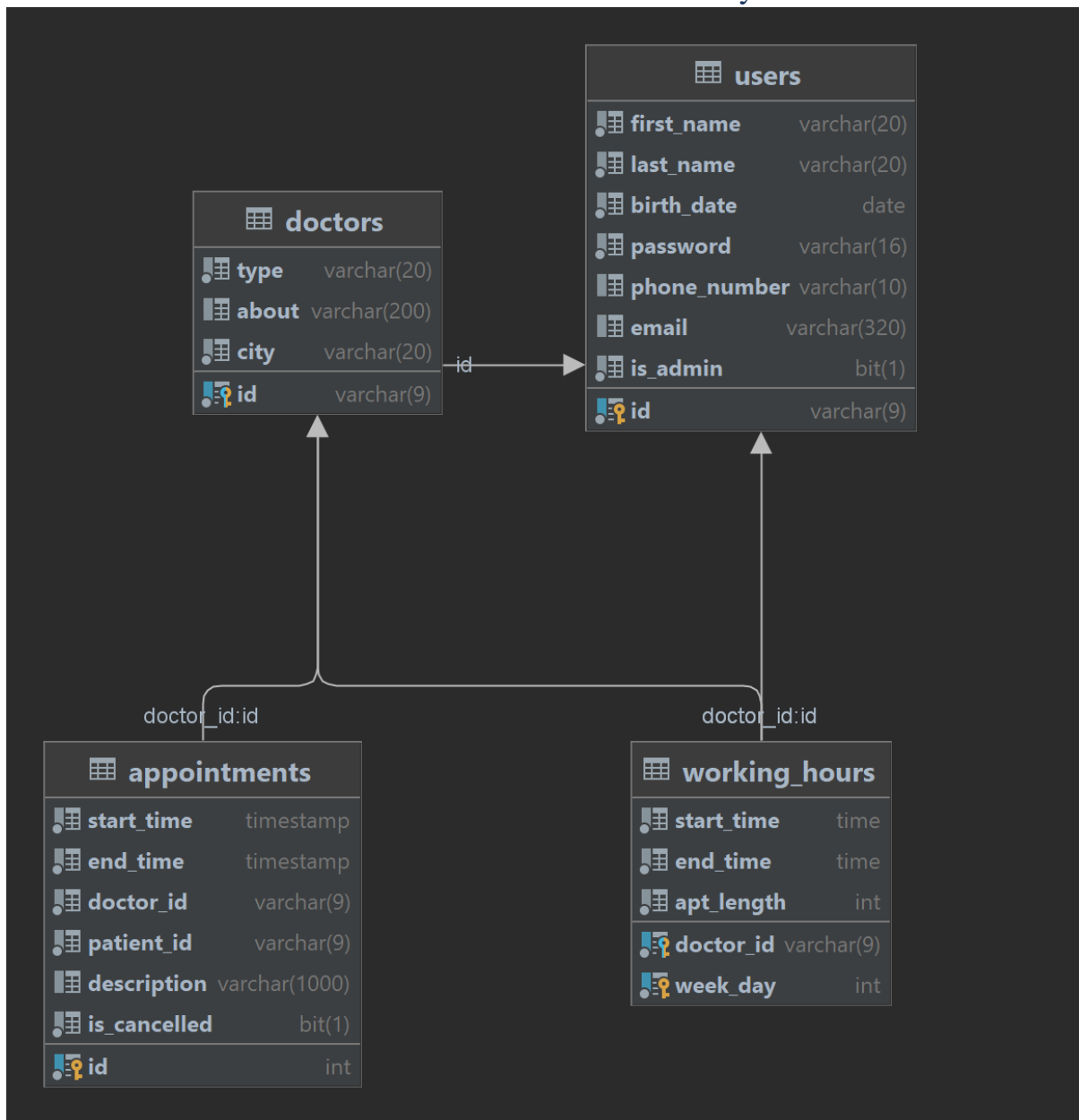
הנחת היסוד על נתוני המערכת בטבלה זו היא ששעות העבודה של רופא בכל שבוע ניתנות על ידי טווח רציף ושהינו קבוע (כלומר אינו משתנה בין שבוע לשבוע, יכול להשתנות בין יום ליום). כמובן שמגבלות אלה ניתן לעקוף בקלות, אך איננו בונים מערכת שמטרתה לתפקד בבית חולים אמיתי. אם כן, המפתח הפומבי של טבלה זאת הינו שני השדות (doctor_id, week_day). עצם עובדה זאת גורר שלכל רופא יהיו לכל היותר 7 שורות בטבלה זו, כמתוכנן.

שם השדה	חתימת השדה
doctor_id	varchar(9) not null
week_day	int not null
start_time	time not null
end_time	time not null
apt_length	int not null default 10
-	primary key (doctor_id, week_day)
-	foreign key (doctor_id) references Doctors(id) on delete cascade
-	foreign key (doctor_id) references Users(id) on delete cascade



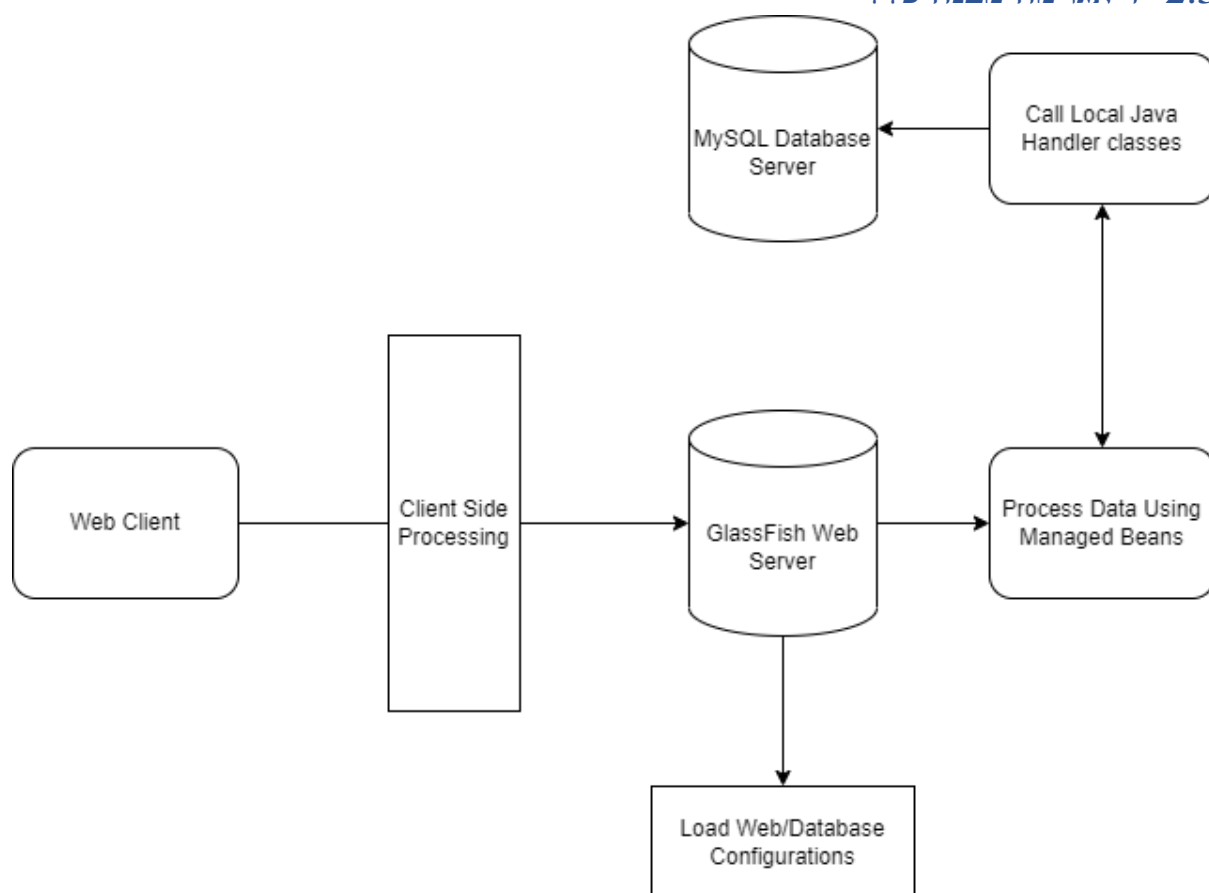


Entity-Relations דיאגרמת 2.2.5



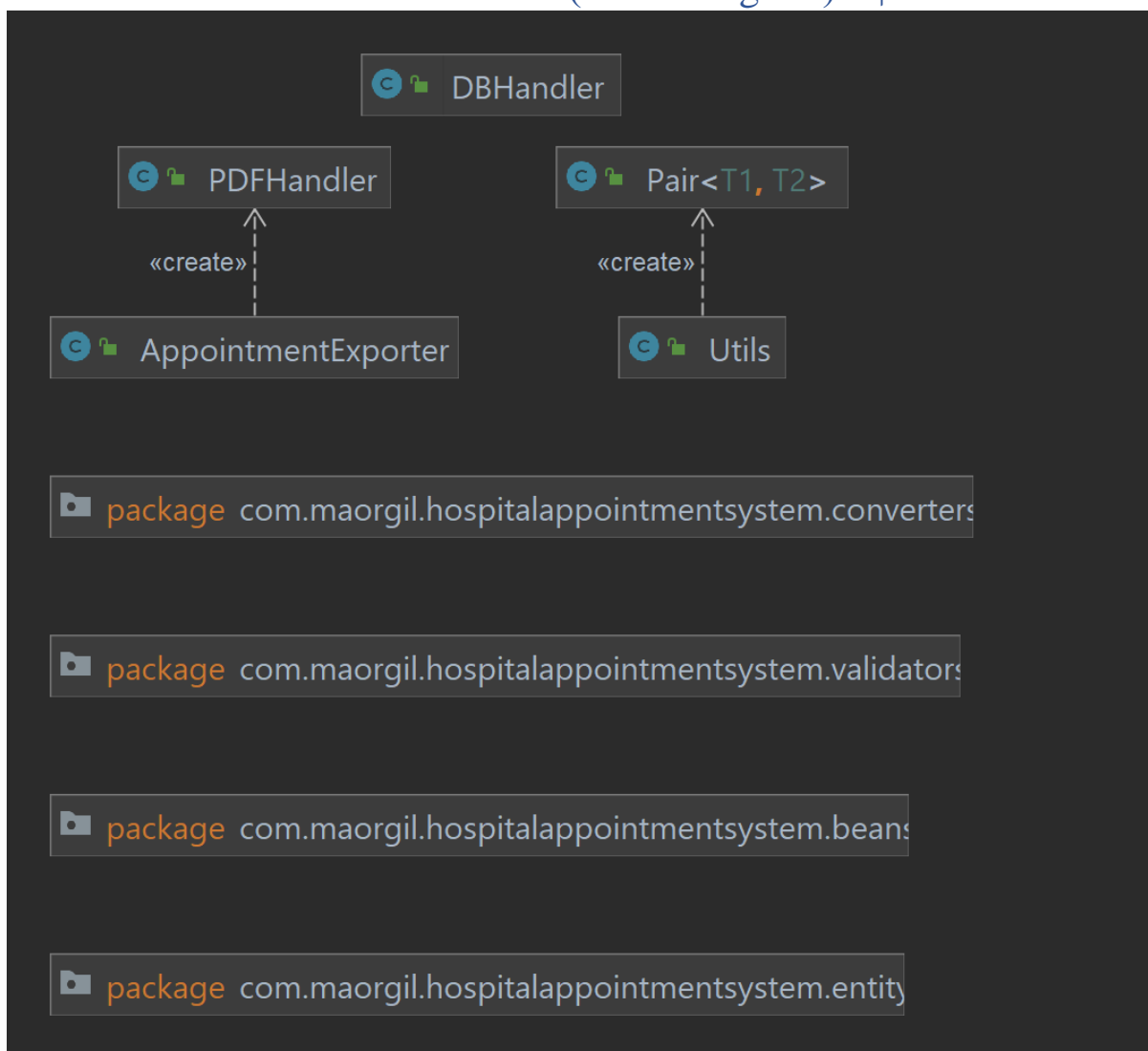


2.3 דיאגרמה מבנה כללי





2.4 דיאגרמת מחלקות (Class Diagram)





3 דפי האתר

בחלק זה נפרט על דפי האתר ומבניהם, או במילים אחרות על קבצי ה-JSF (.xhtml). המרכיבים אותו. עבור כל קובץ שכזה, נסביר את תכליתו ונתאר את המבנה שלו, הרכיבים העיקריים המוגדרים בו.

3.1 דף admin

לדף זה ייעוד מאוד פשוט - לאפשר למשתמש מסוג מנהל (admin) לנהל את בסיס הנתונים של בית החולים. הדף מאפשר הרצת שאילתות SQL ישירות על בסיס הנתונים. הדבר מאפשר למנהל לעשות כל פעולה שירצה, בהסתמך על כך שהוא מכיר את מבנה בסיס הנתונים. דרישה זו אינה פרקטית במיוחד לאתרים "אמיתיים", אך בשל היקף הפרוייקט הסתפקנו בה, מאחר והיא נותנת את הפונקציונליות הדרושה ממנה.

תהינה זאת טרגדיה אם משתמש רגיל יוכל לגלוש לאתר זה ולבצע פעולות כרצונו, ולכן האתר דוחה את הגישה מפני משתמשים רגילים.

רכיב	תיאור
preRenderView-eventListener	הרכיב הינו אלמנט f: event ונמצא באיזור ה- h: head של הקובץ. הרכיב דואג לבדוק שהמשתמש מחובר לפני טעינת הדף ומגרש אותו במידת הצורך.
query-input-panel	הרכיב מורכב מאלמנט p: panel ¹ שמכיל אלמנט p: inputTextArea וקולט את השאילתה להרצה על בסיס הנתונים מהמשתמש.
results-output-panel	הרכיב מורכב מאלמנט p: panel שמכיל אלמנט p: outputLabel ומציג את תוצאות השאילתה לאחר שהורצה על בסיס הנתונים.

3.2 דף appointment-download

הדף משמש להורדת קבצי פגישות למכשירו של המשתמש. בדף הפרופיל ניתן ללחוץ על כפתור הורדה של פגישה מסויימת ובכך הדפדפן יגלוש לכתובת הדף יחד עם פרמטרי הבקשה הנחוצים. מאחר שמזהה הפגישה מועבר בתור פרמטר בקשה, גם בדף זה חשוב ביותר לוודא את זכאות המשתמש ולראות שאכן הפגישה שהתקבלה כפרמטר בקשה אכן שייכת למשתמש המחובר, שכן לא נרצה לאפשר למשתמש מסויים להוריד למכשירו קבצי פגישות של משתמשים אחרים.

רכיב	תיאור
preRenderView-eventListener	הרכיב הינו אלמנט f: event ונמצא באיזור ה- h: head של הקובץ. הרכיב דואג לבדוק שהמשתמש מחובר לפני טעינת הדף ומגרש אותו במידת הצורך.
preRenderView-eventListener	הרכיב גם הוא מאזין לפני טעינת הדף, ומתבצע לאחר פעולת הרכיב הראשון (כלומר לאחר וידוא חיבור המשתמש). הרכיב אחראי לקרוא לפונקציית downloadAppointment הנמצאת ב- appointmentExportBean

3.3 דף appointmentFailure

כאשר משתמש מנסה לשריין פגישה ייתכנו שני תרחישים. הראשון, הבקשה התבצעה בהצלחה והפגישה שוריינה. השנייה, משתמש אחר תפס את הפגישה בזמן שעבר מייצור רשימת הפגישות הזמינות עד שהמשתמש בחר את הפגישה המתאימה לו. התרחשות התרחיש הראשון מתוארת בסעיף הבא, ואילו הסעיף הנוכחי עוסק בתרחיש השני, כשלון.

¹ אלמנטים המתחילים בקידומת 'ק' לקוחים מספריית PrimeFaces ומספקים פונקציונליות ועיצוב Built-In CSS



העמוד פשוט ביותר, ומלבד האלמנטים הנמצאים בכל העמודים הוא מכיל פסקה סטטית (שאינה משתנה) המסבירה למשתמש את המצב.

3.4 דף appointmentSuccess

כפי שצויין בסעיף 3.3 דף appointmentFailure, דף זה מתייחס להצלחת תהליך שיריון פגישה. דף זה מאשר למשתמש כי שריון הפגישה הושלם בהצלחה, ובין היתר מציע לו להוסיף את הפגישה ליומן גוגל שלו (Google Calendar).

רכיב	תיאור
appointment-info	הרכיב מורכב משלושה אלמנטי p:outputLabel המציגים למסך אישור למשתמש שהפגישה נקבעה ואת פרטי הפגישה.
add-to-calendar-button	הרכיב הינו כפתור פשוט שבלחיצתו מנווט את המשתמש בחלון חדש לעמוד יצירת אירוע ב-Google Calendar עם פרטי הפגישה ממולאים (מועברים דרך ה-URL בתור פרמטרי בקשה).

3.5 דף index

דף ה-index הינו דף הבית של האתר. כאשר משתמש מנווט לכתובת האתר הוא מנווט לכתובת index.xhtml. מלבד הרכיבים המופיעים בכל דף, דף הבית מכיל פסקה המספרת מעט על בית החולים (כיתוב גנרי), וכאשר משתמש שאינו מחובר מנווט לדף זה נוסף עבורו כפתור "Sign Up" לתפריט הראשי (בנוסף לאופציה הקיימת תמיד בתפריט הציד).

בנוסף, דף הבית מכיל רשימה המציגה את כל הרופאים של בית החולים והתיאור שלהם, וכאשר עומדים על שם אחד הרופאים עם העכבר מופיעים גם שעות הפעילות שלו והעיר בה עובד.

רכיב	תיאור
signup-button	כפתור פשוט (אלמנט מסוג a עם href) אשר נמצא בקומפוננטת ה-navbar. מוצג רק אם המשתמש אינו מחובר.
description-paragraph	פסקה פשוטה (אלמנט p) המכילה סיפור קצר על בית החולים.
doctors-iterator	הרכיב מורכב מאלמנט c:forEach ² שעובר על רשימת הרופאים ומציג עבור כל אחד אלמנט li עם אלמנט c:forEach נוסף שרץ שעובר על רשימת שעות העבודה השבועיות. קלף שעות העבודה אינו מוצג למסך, אלא כשעומדים על כותרת הרופא.

3.6 דף login

הדף פשוט ביותר ומשמש למטרה אחת בלבד: להתחבר לאתר. באתר זוג תיבות טקסט הנמצאות בתוך טופס המיועדות לקבלת תעודת זהות וסיסמה אשר מאמתות את זהותו של המשתמש.

3.7 דף make-appointment

הדף אחראי לקביעת תורים עבור המשתמש. בטעינת הדף לראשונה, המשתמש ייתקל בטופס המאפשר לו למלא העדפות עבור התור אותו רוצה לאחר. לאחר שישלח את הטופס, העמוד ייטען מחדש וייטען לוח שנה המציג את התורים המוצעים למשתמש בהסתמך על ההעדפות בהם בחר.

² אלמנט c:forEach משתייך לספריית jsp/jstl/core ומאפשר לעבור באופן איטרטיבי על אוסף איברים (Collection) מתוך קובץ ה-index.xhtml.



רכיב	תיאור
preRenderView-eventListener	הרכיב הינו אלמנט f: event ונמצא באיזור ה- h: head של הקובץ. הרכיב דואג לבדוק שהמשתמש מחובר לפני טעינת הדף ומגרש אותו במידת הצורך.
categories-dropdown	רכיב של רשימה לבחירת קטגוריות רופאים (רשימה של בחירת איבר אחד בכל פעם). לאחר בחירת קטגוריה היא תתווסף לרשימת הקטגוריות שנבחרו ותעלם מן הרשימה. הרשימות מתעדכנות בעזרת בקשות ajax.
doctors-dropdown	רכיב של רשימה לבחירת רופאים מסויימים. הרכיב זהה ל- categories-dropdown מלבד לבחירת הרופאים במקום קטגוריות.
locations-dropdown	רכיב של רשימה להגבלת מקומות הפגישה. הרכיב זהה ל- categories-dropdown מלבד לבחירת המיקומים במקום קטגוריות.
from/to-datetime	שתי תוויות תאריכים זהות במימושן. רכיבים מסוג אלמנט p: calendar אשר מעבר להקליד תאריך ושעה בפורמט המוזן או לבחור מלוח שנה הקובץ בעת לחיצה על תווית הטקסט.
submit-button	כפתור p: commandButton פשוט לשליחת הטופס וקבלת תורים מוצעים.
event-schedule	טופס ובתוכו אלמנט מרכזי מסוג p: schedule המאפשר טעינת אירועים ומתן תגובה ללחיצה עליהם מ- managed bean שיוצג בהמשך.
back-button	כפתור p: commandButton פשוט לחזרה אחורה לטופס ההעדפות. ההעדפות נשמרות בחזרה אחורה.
reserve-dialog	אלמנט מסוג p: dialog אשר מוגדר בגוף קובץ ה- xhtml, ובזכות התכונה widgetVar מתאפשר להציגו למסך מ- managed bean ומקוד JS שיוצג בהמשך. הדיאלוג מופיע למרכז המסך בלחיצה על תור, מציג את פרטיו (ע"י בקשת ajax כך שאין צורך בטעינת הדף מחדש) ומציע למשתמש לשריין את התור שנלחץ (ומשם מועבר המשתמש להסתעפות המתוארת בסעיף 3.3).

3.8 דף myprofile

לדף מספר שימושים רבים, נתאר את כולם לפי סדר הופעתם באתר.

ראשית, בראש הדף מופיע form עם שדות טקסט המציגים את פרטי המשתמש (שם, תאריך לידה, טלפון, מייל, ועוד...). בתחתית ה- form נמצא כפתור דינמי – כאשר ה- form במצב "עריכה" הכפתור מציג "Submit" וכאשר ה- form במצב תצוגה הכפתור מציג "Edit". במצב "תצוגה" לא ניתן לערוך את השדות, והם מופיעים בתצוגת "readonly".

שנית, לאחר חלק המידע של המשתמש, מופיע איזור בו המשתמש יכול לצפות בכל הפגישות שלו (בעבר, הווה, עתיד, ואלה שבוטלו). מצד שמאל מופיע תפריט בעזרתו המשתמש יכול לסנן את הפגישות שברצונו לראות (ע"י רופא, טווח תאריכים, פגישות שלא בוטלו, ועוד...) ולמיון אותן לפי מאפיינים כגון זמן התחלה או שם הרופא. בכל אחד מכרטיסי הפגישות, מופיע כפתור הורדה (download) שבלחיצה עליו מוריד את קובץ הפגישה למכשירו של המשתמש, ובמידה והפגישה אינה מבוטלת מופיע כפתור נוסף שבלחיצתו המשתמש יוכל לבטל את הפגישה.

בנוסף, אם המשתמש הינו רופא (הרי רופאים מיוצגים באתר בתור משתמשים גם הם) יופיע חלק נוסף, המציג לרופא את הפגישות הקרובות שלו (בהן הוא רופא). לרופא יש כפתור עריכה בקלף הפגישה ויוכל לשנות את תיאור הפגישה.



רכיב	תיאור
preRenderView-eventListener	הרכיב הינו אלמנט f:event ונמצא באיזור ה- h:head של הקובץ. הרכיב דואג לבדוק שהמשתמש מחובר לפני טעינת הדף ומגרש אותו במידת הצורך.
user-data-grid	הרכיב הינו טבלה עם 3 עמודות (h:panelGrid) המכיל את נתוני השדות של המשתמש. העמודה הראשונה שמורה לרכיב h:outputText אשר מתאר את מידע השדה (למשל, שם פרטי). העמודה השנייה שמורה לשדה עצמו. העמודה השלישית שמורה להודעות שגיאה של רכיבי validator, למשל (פורמט לא נכון של כתובת מייל או מספר טלפון בן ספרה אחת).
edit-button	כפתור שעובר למצב עריכה של השדות ברכיב הקודם. הכפתור מרונדר (לפי תכונת rendered) רק אם הטופס במצב "תצוגה".
submit-button	כפתור שעובר למצב תצוגה ושולח את הנתונים לבסיס הנתונים. הכפתור מרונדר (לפי תכונת rendered) רק אם הטופס במצב "עריכה".
filters-container	טופס אשר מכיל מספר מאפיינים אשר לפיהם ניתן לסנן פגישות. בתחתית מופיע כפתור "Search" אשר בלחיצתו מפעיל קריאה חדשה לבסיס הנתונים וטוען מחדש את הדף.
from/to-date-filter	שתי תוויות תאריכים מסוג h:inputText אשר קוראות תאריכים ומגבילות את טווח התאריכים עבור הפגישות המוצגות.
doctor-filter	רכיב מסוג h:selectOneMenu אשר מציג את כל הרופאים אשר מטפלים במשתמש (לא כל הרופאים של בית החולים) ומציע למשתמש לבחור אחד מהם (או את כולם). טעינת הרופאים מתבצעת באמצעות רכיב f:selectItems ³ .
sort-by-menu	רכיב מסוג h:selectOneMenu המציג 3 אופציות פשוטות: "Date", "Doctor", "Length", כלומר לפי תאריך הפגישה, לפי שם הרופא (א"ב) או לפי אורך הפגישה. כאן האופציות נטענות בעזרת 3 רכיבי f:selectItem.
asc/desc-switch	כפתור בוליאני מסוג switch שקובע האם לסדר את הפגישות בסדר עולה או יורד (לפי המיון שנקבע ברכיב sort-by-menu). משתמשים ב- ajax על מנת לעדכן את הטקסט המלווה את הכפתור בזמן אמת.
include-cancelled	כפתור בוליאני מסוג h:selectBooleanCheckbox אשר קובע האם לכלול ברשימת הפגישות פגישות שבוטלו.
appointments-card-list	רשימה מסודרת של קלפי פגישות. הרשימה ממומשת באופן איטרטיבי על ידי אלמנט ui:repeat ⁴ . כל קלף ברשימה מכיל כותרת פגישה (עבור מטופל) ואת התיאור שלה בקטן. פגישות מבוטלות או שהיו בעבר מעוצבות בסגנון מתאים. בתחתית כל קלף מופיעה תיבת flex שמסדרת את הכפתורים של אותו קלף. אם ניתן לבטל את הפגישה, מופיע כפתור ביטול (אשר מקפיץ דיאלוג אישור). עבור כל הפגישות מופיע כפתור הורדה שבלחיצה עליו המשתמש מנווט לעמוד appointment-download המתואר בסעיף 3.2 עם ערך פרמטר הבקשה id המתאים (מזהה הפגישה).
appointment-cancel-dialog	אלמנט p:dialog אשר קופץ למסך בעת לחיצה על אחד מכפתורי ביטול הפגישה ומאשר את ביטול הפגישה עם המשתמש. הדיאלוג מקבל widgetVar ומופעל מ-managed bean שיתואר בהמשך.
doctors-appointments-card-list	רכיב זה דומה במקצת לרכיב appointments-card-list. הרכיב ממומש באופן איטרטיבי בעזרת אלמנט c:forEach. בניגוד לרכיב appointments-card-list, כותרת הפגישות היא עבור הרופא (כלומר מכילה את שם המטופל במקום שם הרופא), והכפתור היחיד שמופיע בתיבת ה-flex הינו כפתור עריכת תיאור הפגישה. כמו כן, לא קיימת דרך לסנן את הפגישות, והרשימה מכילה את 5 (במקסימום) הפגישות הקרובות (עתידיות) של הרופא שאינן מבוטלות.

³ בדומה לרכיבי c:forEach ו- ui:repeat, f:selectItems מקבל Collection וממיר אותו לרכיבים בתפריט. ניתן לקבוע ערך מידעי עבור הרכיבים (כמו למשל ID) וערך טקסטואלי שונה לחלוטין שיוצג בתפריט.

⁴ אלמנט ui:repeat דומה מאוד לאלמנט c:forEach, אך משתייך לספריית primefaces/ui ומציע פונקציונליות יחודית.



<p>אלמנט <code>p: dialog</code> אשר קופץ למסך בעת לחיצה על אחד מכפתורי עריכת התיאור שברשימה של הרופא. תיבת הטקסט שבדיאלוג תכיל את תיאור הפגישה שנחלצה. לאחר שינוי התיאור ולחיצה על כפתור האישור תתבצע בקשת <code>ajax</code> שתעדכן את כרטיס הפגישה אצל הרופא. (נשים לב שאין צורך לעדכן כרטיס ברשימת המשתמש כי לא ייתכן רופא יהיה מטופל אצל עצמו, או במילים אחרות $doctor_id \neq patient_id$), ולכן אף פגישה לא תופיע פעמיים.</p>	<p>description-edit-dialog</p>
---	--------------------------------

3.9 דף signup

הדף מכיל את שדות האינפורמציה של משתמש הנמצאים בבסיס הנתונים. משתמש שנרשם חייב למלא פרטים מסויים, אולם את כתובת המייל שלו או את מספר הטלפון, למשל, אינו מוכרח למלא כדי להשלים את ההרשמה (ותמיד יוכל להשלימם בעמוד הפרופיל).

בין היתר, הדף מדגים שימוש באלמנטי ה- `validator` אשר מאפשר וידוא תקינות פורמט המידע בצד המשתמש (לוקאלית).

רכיב	תיאור
input-fields-grid	<p>בדומה לאלמנט ה- <code>h: panelGrid</code> המופיע בראש עמוד הפרופיל, אלמנט זה מכיל גם הוא את שדות המשתמש (אך רק אלה שיכול להכניס בעצמו, למשל לא מכיל את שדה ה- <code>user-type</code>). בעמוד הפרופיל יכול לשנות המשתמש רק את כתובת המייל שלו ואת מספר הטלפון, כלומר לא יכול לשנות, למשל את שמו. פרטים אלה מתקבלים כקלט מן המשתמש פעם אחת בלבד, בעת ההרשמה. בדומה לעמוד הפרופיל, גם אלמנט זה הינו טבלה בעלת 3 עמודות באותו מבנה, כאשר פה השימוש ב- <code>validators</code> מעט כבד יותר, וממחיש אותם בצורה מצויינת (למשל, תעודת זהות צריכה להתאים לביטוי רגולרי מהצורה $[0 - 9]^9$).</p>
submit-fields	<p>הרכיב מכיל שני אלמנטים. הראשון, כפתור <code>h: commandButton</code> רגיל, אשר מוודא בצד השרת את ה- <code>validators</code> לפני שליחת הטופס ומגיב בהתאם. השני, תיבת טקסט מסוג <code>h: outputText</code> אשר מציגה את סוג השגיאה בחזרה מן השרת במידה ויש כזו (למשל, משתמש עם תעודת זהות כבר קיים).</p>





4 תוספות Web

מלבד הקוד וקבצי הדפים (xhtml), הפרויקט כולל מספר מרכיבים נוספים הקשורים לבניית Web Application.

4.1 קבצי קונפיגורציה XML

ישנם מספר קבצי קונפיגורציה גנריים אשר עוסקים בטכנולוגיית ה JSF, JPA, ועוד. הרשימה כוללת את הקבצים:

- beans.xml – קונפיגורציות של managed beans, קובץ דיפולטיבי.
 - faces-config.xml – קונפיגורציות של JSF, קובץ דיפולטיבי.
 - web.xml – קונפיגורציות של JSF, הגדרות האתר (למשל, welcome-file).
 - persistence.xml – קובץ קונפיגורציה של JPA המקשר את בסיס הנתונים, הגדרות, סיסמאות וכו'.
 - pom.xml – קובץ קונפיגורציה של Maven, כולל בין היתר את ה- dependencies של הפרויקט.
- מלבד הקבצים הללו, קיימים קבצי JS ו-CSS שמהווים חלק בלתי נפרד מן הפרויקט. לא נקדיש להם פרק משלהם, אך שווה לציין את הקבצים השונים.

4.2 קבצי JS

בהתאם לתוכן הקורס, מיקוד הפרויקט הוא בכתיבת קוד Java, אך במקרים מסויימים קל יותר לבצע פעולות באמצעות פקודת JavaScript בודדה במקום שורות קוד ב-Java, בייחוד כשמתמשים בספריות כגון PrimeFaces. פונקציות העזר מפוצלות לשני קבצי JS:

- קובץ profilepage.js – מיועד לדף הפרופיל.
- קובץ sidebar.js – מטפל בפעולות ה- sidebar ולכן נכלל בכל הדפים.

4.3 קבצי CSS

קבצי CSS עוסקים בעיצוב דפי האתר, וסידור קומפוננטות ה- html כרצוי. הקבצים אינם כוללים קוד משמעותי אלא תכונות עיצוביות ואינם מעניינים מידי, ולכן לא נקדיש יותר מלציון את שמות הקבצים ומשמעותם:

- styles.css – קובץ המכיל מחלקות CSS המשומשות במספר קבצים. המחלקות בקובץ זה כלליות יותר, וכתובת בצורה שיכולה להתאים ליותר מקומות. הקובץ נכלל ע"י @import בכל שאר קבצי ה-CSS.
- home.css – עיצוב ספציפי לדף הבית (index.xhtml).
- make-appointment.css – עיצוב ספציפי לדף יצירת הפגישה (make-appointment.xhtml).
- profilepage.css – עיצוב ספציפי לדף הפרופיל (myprofile.xhtml).

נשים לב שלרוב הדפים לא קיים קובץ CSS משלהם, מכיוון שלא קיימות בהם מספיק דרישות עיצוביות שדורשות (לדעתנו, הרי מדובר בבחירה ארגונית) ארגון של העיצוב בקובץ חיצוני, אלא ניתן להסתפק בהגדרת התכונות הפשוטות בעזרת תכונת style, או שימוש במחלקות עיצוב הקיימות בקובץ styles.css.

4.4 קבצי Assets

סוג נוסף של קבצים הינם קבצי מדיה. במקומות מסויימים לא הסתפקו בקבצים אשר מסופקים בתור ברירת המחדל על ידי הספריות השונות, ונאלצו להוריד קבצי מדיה מן האינטרנט או ליצור כאלה בעצמם. קבצי מדיה אלה נמצאים בתקיית assets הכוללת את הקבצים:

profile-icon.png, pencil-icon.png, favicon.ico, download-icon.png, cancel-icon.png





5 מחלקות JSF's Managed Bean

המחלקות המסומנות ב- @ManagedBean Annotation מיועדות לספק ניהול backend לדפי האתר. העמודים יכולים לציין משתנים ושדות המחוברים ל- managed bean כלשהו ובכך לקבל משתנים הנקבעים על ידי קוד Java רגיל. מחלקות ה- managed beans בפרויקט מוגדרות במארו (package) beans שנמצא בתיקייה שנוצרת בבירור המחולל של פרוייקט ה- JSF: "java". נתאר כעת את מחלקות האפונים והתוכן החשוב בהן:

5.1 מחלקת AdminBean

המחלקה מנהלת את דף המנהל. באופן כללי, המחלקה מקשרת את הדף לבסיס הנתונים, מבצעת שאילתות שמקבלת קלט מן המשתמש על בסיס הנתונים, ומציגה בחזרה את הקלט.

סוג מאפיין	שם/חתימה	תיאור
Class Annotation	ManagedBean	הצהרה על המחלקה בתור managed bean ומתן שם.
Instance Variable	String query	שדה מוצמד (Linked Field ⁵) עבור קלט השאילתה
Instance Variable	String result	משתנה אשר מאחסן את תוצאת השאילתה ומציג אותה לדף בעזרת פונקציית .get
Function	void execute()	פונקצייה אשר מריצה את השאילתה על בסיס הנתונים, ומאחסנת את התוצאה במשתנה .result

5.2 מחלקת AppointmentExportBean

המחלקה מטפלת בהורדת קובץ פגישה למכשירו של המשתמש ונקראת מדף appointment-download.

סוג מאפיין	שם/חתימה	תיאור
Class Annotation	ManagedBean	הצהרה על המחלקה בתור managed bean ומתן שם.
Function	String isUserEligible(AppointmentsEntity)	פונקצייה שבודקת האם המשתמש המחובר זכאי להוריד את הפגישה המבוקשת (דרך משתנה בקשה).
Function	StreamedContent downloadAppointment	פונקצייה אשר טוענת את הקובץ ויוצרת את המענה המכיל את הקובץ.

5.3 מחלקת AppointmentFormBean

בכניסה לדף make-appointment יש לבחור העדפות לפגישה ולמלאן בטופס. מחלקה זאת עומדת מאחורי פונקציונליות זאת.

סוג מאפיין	שם/חתימה	תיאור
Class Annotation	ManagedBean	הצהרה על המחלקה בתור managed bean ומתן שם.
Class Annotation	SessionScoped	ה- bean הינו SessionScoped על מנת שהעדפות המשתמש ייזכרו בין אם מנווט בין דפים.
Interface Implementation	Serializable	הספריות דורשות לעיתים מן ה- beans לממש ממשק זה (תלוי באילו רכיבים משתמשים).
Instance Variable	boolean submitted	מאחר והדף בעל שני מצבים (לפני שליחת הטופס ואחריו), יש לשמור את מצב המשתמש בזרימת המידע.

⁵ שדה מוצמד (Linked Field) הינו משתנה מופע של Managed Bean כלשהו, אשר מוצמד לשדה קלט כלשהו מן המשתמש ע"י פונקציות .get/set.



שדה מוצמד לשדה ה- from בדף.	LocalDateTime fromTime	Instance Variable
שדה מוצמד לשדה ה- to בדף.	LocalDateTime toTime	Instance Variable
שדה מוצמד לשדה תפריט המקומות בדף.	String selectedLocation	Instance Variable
רשימה של המקומות בהם ניתן לבחור (עליה עובר הרכיב האיטרטיבי בתפריט chooseOne).	List<String> locations	Instance Variable
רשימה של המקומות שנבחרו עד כה (עליה עובר הרכיב האיטרטיבי ברכיב dataList).	List<String> selectedLocations	Instance Variable
שדה מוצמד לשדה תפריט הרופאים בדף.	String selectedDoctor	Instance Variable
רשימה של הרופאים אותם ניתן לבחור (עליה עובר הרכיב האיטרטיבי בתפריט chooseOne). כאן הרופאים מיוצגים ע"י ה- id שלהם (מפתח ראשי).	List<String> doctors	Instance Variable
רשימה של המקומות שנבחרו עד כה (עליה עובר הרכיב האיטרטיבי ברכיב dataList).	List<String> selectedDoctors	Instance Variable
שדה מוצמד לשדה תפריט הקטגוריות בדף.	String selectedCategory	Instance Variable
רשימה של הקטגוריות בהם ניתן לבחור (עליה עובר הרכיב האיטרטיבי בתפריט chooseOne).	List<String> categories	Instance Variable
רשימה של הקטגוריות שנבחרו עד כה (עליה עובר הרכיב האיטרטיבי ברכיב dataList).	List<String> selectedCategories	Instance Variable
שדה מוצמד ל- switch ה- categories/doctors-by.	boolean byCategory	Instance Variable
מחלקה המחזיקה את השדות הדרושים לאלגוריתם יצור רשימת התורים. מחלקה נפרדת על מנת לשמור על עקרון הצמידות הנמוכה .	FormResults	Static Inner Class
עדכן את ה- FormResults האחרונים ושלא את הקובץ, סמן submitted=true וטען מחדש את הדף.	String submit()	Function

5.4 מחלקת AppointmentsCalendarBean

המחלקה טוענת ומנווטת את לוח השנה לבחירת התור, ונטענת לאחר פעולת ה- bean המתואר בסעיף 5.3.

סוג מאפיין	שם/חתימה	תיאור
Class Annotation	ManagedBean	הצהרה על המחלקה בתור managed bean ומתן שם.
Class Annotation	ViewScoped	על מנהל לוח השנה להיות ViewScoped על מנת שייטען מחדש מבסיס הנתונים לאחר שהתבצע Back והנתונים ישתנו.
Interface Implementation	Serializable	הספריות דורשות לעיתים מן ה- beans לממש ממשק זה (תלוי באילו רכיבים משתמשים).
Instance Variable	ScheduleModel eventModel	הרכיב מהווה את המודל שרץ מאחורי הלקעים של אלמנט ה- p: schedule. נטען דרך ה- bean ומועבר דרך get לדף.
Instance Variable	AppointmentsEntity lastClickedAppointment	שמירת הפגישה שנלחצה למען הדיאלוג שבעל View שמשמש ב- context אחר.
Function	void init()	אתחול לוח השנה עם הפגישות מהרצת האלגוריתם.
Function	List<AppointmentsEntity> genAppointments(FormResults)	הפונקציה העילית ביותר באלגוריתם יצור הפגישות.





Function	void OnEventSelect(SelectEvent)	עדכן את נתוני הפגישה שנבחרה ב-bean, שלח בקשת ajax לאלמנט הדיאלוג, והקפץ אותו.
Function	void reserve()	קבע את הפגישה באופן סופי. שמור לבסיס הנתונים ונווט לדף הצלחה/כשלון.

5.5 מחלקת DoctorListBean

המחלקה עוסקת בהבאת רשימה של רופאים וחיפושם בבסיס הנתונים לפי הדרישות בדף הבית.

סוג מאפיין	שם/חתימה	תיאור
Class Annotation	ManagedBean	הצהרה על המחלקה בתור managed bean ומתן שם.
Instance Variable	String searchTerm	שדה מוצמד לשדה החיפוש עבור שמות הרופאים.
Function	List<DoctorsEntity> getDoctors()	מחפשת את הרופאים בבסיס הנתונים על פי ערך החיפוש ומחזירה רשימה של רופאים מתאימים לרכיב האיטרטיבי.
Function	List<WorkingHoursEntity> getWorkingHours(String)	מחפשת בבסיס הנתונים את שעות העבודה של הרופא הנתון (המחרוזת מייצגת תעודת זהות של רופא) ומחזירה רשימה של שעות עבודה לרכיב האיטרטיבי.

5.6 מחלקת LoginBean

המחלקה עוסקת בקבלת בקשות התחברות של משתמשים ושמירת פרטי המשתמש המחובר למשך הסשן.

סוג מאפיין	שם/חתימה	תיאור
Class Annotation	ManagedBean	הצהרה על המחלקה בתור managed bean ומתן שם.
Class Annotation	SessionScoped	כמוסבר, יש לשמור את פרטי המשתמש המחובר למשך כל ה-Session
Instance Variable	String id	שדה מוצמד לשדה קלט תעודת הזהות
Instance Variable	String password	שדה מוצמד לשדה קלט הסיסמה
Instance Variable	String output	שדה המוצג בתיבת טקסט להצגת השגיאה המתאימה (סיסמה לא נכונה וכו').
Instance Variable	boolean loggedIn	שדה המשמש את כל ה-beans לדעת האם משתמש מסוים מחובר או לא (ע"י get כמובן).
Function	String submit()	בדיקת הקלט מן המשתמש עם בסיס הנתונים וקידומו בהתאמה.
Function	String logout()	נתק את המשתמש, קטע את הסשן, ונווט לעמוד הבית.

5.7 מחלקת ProfileAppointmentsBean

המחלקה מגבה את חלק פגישות המשתמש (בתור מטופל) והסינון שלהן בעמוד הפרופיל.

סוג מאפיין	שם/חתימה	תיאור
Class Annotation	ManagedBean	הצהרה על המחלקה בתור managed bean ומתן שם.



דריסת הבנאי הדיפולטיבי לשם התחברות לבסיס הנתונים וקבלת רשימת הפגישות עם סינון ברירת מחדל.	ProfileAppointmentsBean()	Constructor
שדה מוצמד לשדה קלט סינון תאריך מינימלי.	Date fromDate	Instance Variable
שדה מוצמד לשדה קלט סינון תאריך מקסימלי.	Date toDate	Instance Variable
רשימה של כל הפגישות החוזרת מחיפוש בבסיס הנתונים, לאחר סינון. מועברת לרכיב האיטרטיבי.	List<AppointmentsEntity> appointments	Instance Variable
שדה מוצמד לשדה התפריט – בחירת סוג מיון.	String sortBy	Instance Variable
שדה מוצמד לכפתור הבוליאני של כיוון הסדר (עולה/יורד).	boolean isAscending	Instance Variable
שדה מוצמד לתיבת ה- checkbox של הכללת פגישות שבוטלו.	boolean includeCancelled	Instance Variable
נקראת בלחיצת על כפתור הסינון – קריאה לבסיס הנתונים ושמירת הרשימה ברשימת appointments	void search()	Function
נקראת בלחיצה על כפתור הביטול של אחת מהפגישות, שמירת הפגישה שסומנה, שליחת בקשת ajax לעדכון הדיאלוג והקפצתו.	void onCancelAppointment (AppointmentsEntity app)	Function
נקראת מכפתור הדיאלוג לאישור – סימון is_cancelled=true בבסיס הנתונים וטעינת הדף מחדש.	void cancelAppointment()	Function

5.8 מחלקת ProfileDataBean

המחלקה עוסקת בניהול החלק של מידע המשתמש (השדות השמורים בבסיס הנתונים) בעמוד הפרופיל.

סוג מאפיין	שם/חתימה	תיאור
Class Annotation	ManagedBean	הצהרה על המחלקה בתור managed bean ומתן שם.
Class Annotation	SessionScoped	שמירת הפרטים שהמשתמש משנה בין בקשות.
Instance Variable	UsersEntity user	שמירת המשמש המחובר, שמירת השדות, אין צורך לפנות לבסיס הנתונים עם תז כל פעם.
Instance Variable	boolean previewMode	האם במצב "עריכה" או במצב "תצוגה"? לפי ערך משתנה זה משתנה רינדור (rendering) הדף.
Function	String submit()	שלח את הטופס (לאחר שבדקו רכיבי ה- validator לוקאליזציה), שמור את השינויים לבסיס הנתונים, טען מחדש את הדף.
Function	String edit()	סמן previewMode=true ועבור למצב עריכה.

5.9 מחלקת ProfileDoctorBean

מחלקה זו עוסקת בחלק של הפגישות הקרובות של הרופא בתחתית עמוד הפרופיל.

סוג מאפיין	שם/חתימה	תיאור
Class Annotation	ManagedBean	הצהרה על המחלקה בתור managed bean ומתן שם.
Instance Variable	String description	שדה מוצמד לשדה קלט/פלט תיאור הפגישה בדיאלוג הנפתח בעת לחיצה על כפתור העריכה בפגישה מסויימת.



נקראת בעת לחיצה על כפתור עריכת תיאור פגישה כלשהי. שמירת הפגישה, עדכון הדיאלוג באמצעות בקשת ajax והקפצתו למסך.	void onEditDesc (AppointmentsEntity)	Function
נקראת בעת לחיצה על כפתור ה- Save שבדיאלוג. שומרת את התיאור החדש לבסיס הנתונים ומעדכנת את הפגישה שתיאורה נערך באמצעות בקשת ajax (ללא טעינת הדף מחדש!).	void saveDescription()	Function

5.10 מחלקת SignupBean

המחלקה עוסקת בניהול עמוד ה- signup ורישום המשתמש בבסיס הנתונים.

סוג מאפיין	שם/חתימה	תיאור
Class Annotation	ManagedBean	הצהרה על המחלקה בתור managed bean ומתן שם.
Instance Variable	String id	שדה מוצמד לשדה קלט תעודת הזהות
Instance Variable	String firstName	שדה מוצמד לשדה קלט השם הפרטי
Instance Variables	-	שדות מוצמדים זהים עבור lastName, password, birthDate, email, phoneNumber
Instance Variable	String output	שדה שמחזיק את השגיאה אם קיימת כזאת, מוצג למסך בשדה השגיאה בהתרחשות תקלה.
Function	String submit()	ניסיון לרישום המשתמש בבסיס הנתונים – החזרת כשלוך+הודעת שגיאה/הצלחה בהתם לתוצאה.

5.11 מחלקת SuccessAppointmentBean

המחלקה מטפלת בהצגת הפרטים של הבקשה ששורינה בהצלחה בעמוד יצירת הפגישה (make-appointment.xhtml). כמו כן, מפנה את המשתמש ל- URL הנכון עם פרמטרי הבקשה הנכונים בלחיצה על כפתור "Add to Google Calendar".

סוג מאפיין	שם/חתימה	תיאור
Class Annotation	ManagedBean	הצהרה על המחלקה בתור managed bean ומתן שם.
Class Annotation	View Scoped	המחלקה צריכה להיבנות מחדש בכל טעינת מצב חדש על מנת לטעון את נתוני הפגישה הנכונים והמעודכנים ביותר בבנאי.
Constructor	SuccessAppointmentBean()	דריסת הבנאי הדיפולטיבי – טעינת הפגישה האחרונה שנקבעה עבור המשתמש המחובר.
Function	void addToGoogleCalendar()	נקראת בעת לחיצה על הכפתור המתואר, יצירת ה- URL המתאים לפי הדוקומנטציה של גוגל וניווט עם פרמטרי בקשה. הבהרה: אין אין מתעסקים עם ה- API של Google או בידוא משתמשים.

5.12 מחלקת UserRedirectionBean

המחלקה משומשת במספר דפים ומקומות. הפונקציות שבה משמשות כדי לדעת את סטטוס המשתמש ממערכת: מחובר או לא, מנהל או לא..

סוג מאפיין	שם/חתימה	תיאור
------------	----------	-------





הצהרה על המחלקה בתור managed bean ומתן שם.	ManagedBean	Class Annotation
טעינה מחדש בכל קריאה לפונקציה (כי נדרש המידע העדכני ביותר)	ViewScoped	Class Annotation
בודק עם ה- LoginBean האם המשתמש מחובר ומנווט לעמוד ה- login במידה ולא.	void checkLoggedIn	Function
בודק עם ה- LoginBean האם המשתמש מנהל ומנווט לעמוד ה- login במידה ולא.	void checkAdmin	Function





6 מחלקות JPA's Entity

טכנולוגיית JPA מאפשרת לנו גישה נוחה ביותר לאובייקטים המקושרים לבסיס הנתונים, והיא דורשת שנגדירם בצורה מסויימת. בכל אחת ממחלקות ה- Entity (או EntityPK) מופיעים השדות המופיעים בטבלאות בסיס הנתונים עם ה- Annotations המתאימים (למשל, מפתח ראשי יותייג ב- Annotation @Id).

במקום לציין פעם אחר פעם את מבני הנתונים המקבילים למבני הנתונים השמורים ישירות בבסיס הנתונים, נצייןם כאן בטבלה, ולאחר מכן נפרט מעט על כל מחלקה בפרט.

Type בבסיס הנתונים	Type במחלקת Entity
INT	int
VARCHAR(?)	String
BIT	boolean
TIMESTAMP	Timestamp (java.sql) – expands to LocalDateTime
DATE	Date (java.sql) – expands to LocalDate
TIME	Time (java.sql) – expands to LocalTime

כל מחלקה מכילה את כל השדות (עמודות) של הטבלה המתאימה בבסיס הנתונים, כאשר השדה במחלקה מתוייג בתור @Column(name=column_name). בנוסף, כל המחלקות מתוייגות ב- @Entity Annotation (הריק) וב- @Table(name=table_name, schema="hospital").

6.1 מחלקת AppointmentsEntity

סוג מאפיין	שם/חתימה	תיאור
Instance Variable Annotation	@Id	הצהרה על משתנה המופע id בתור המפתח הראשי של הטבלה.
Function	boolean equals(Object)	דריסת equals, השוואה בין אובייקטים לפי השוואת כל השדות.
Function	int hashCode()	דריסת hashCode, חישוב hash value נכון בהתאם לשדות.

6.2 מחלקת DoctorsEntity

סוג מאפיין	שם/חתימה	תיאור
Instance Variable Annotation	@Id	הצהרה על משתנה המופע id בתור המפתח הראשי של הטבלה.
Function	boolean equals(Object)	דריסת equals, השוואה בין אובייקטים לפי השוואת כל השדות.
Function	int hashCode()	דריסת hashCode, חישוב hash value נכון בהתאם לשדות.
Function	String toString()	דריסת toString לנוחות הצגת שם הרופא (דרך קריאה לבסיס הנתונים כי הטבלה לא מכילה שם אלא רק Foreign Key).

6.3 מחלקת UsersEntity

סוג מאפיין	שם/חתימה	תיאור
Instance Variable Annotation	@Id	הצהרה על משתנה המופע id בתור המפתח הראשי של הטבלה.
Function	boolean equals(Object)	דריסת equals, השוואה בין אובייקטים לפי השוואת כל השדות.



Function	int hashCode()	דריסת hashCode, חישוב hash value נכון בהתאם לשדות.
----------	----------------	--

6.4 מחלקת WorkingHoursEntity

סוג מאפיין	שם/חתימה	תיאור
Class Annotation	@IdClass	המפתח הראשי מורכב מכמה שדות ולכן משתמשים במחלקה (מין struct) כדי לייצג אותו.
Instance Variable Annotation	@Id	הצהרה על משתנה המופע doctorId בתור המפתח הראשי (אחד מן השדות) של הטבלה.
Instance Variable Annotation	@Id	הצהרה על משתנה המופע weekDay בתור המפתח הראשי (אחד מן השדות) של הטבלה.
Function	String toString()	דריסת toString, הצגת היום בשבוע ושעות העבודה.
Function	boolean equals(Object)	דריסת equals, השוואה בין אובייקטים לפי השוואת כל השדות.
Function	int hashCode()	דריסת hashCode, חישוב hash value נכון בהתאם לשדות.

6.5 מחלקת WorkingHoursEntityPK

סוג מאפיין	שם/חתימה	תיאור
Instance Variable Annotation	@Id	הצהרה על משתנה המופע doctorId בתור המפתח הראשי (אחד מן השדות) של הטבלה.
Instance Variable Annotation	@Id	הצהרה על משתנה המופע weekDay בתור המפתח הראשי (אחד מן השדות) של הטבלה.
Interface Implementation	Serializable	מחלקות מפתח צריכות לממש את הממשק כדי להשוות מפתחות לפי pure binary.
Function	boolean equals(Object)	דריסת equals, השוואה בין אובייקטים לפי השוואת כל השדות.
Function	int hashCode()	דריסת hashCode, חישוב hash value נכון בהתאם לשדות.





7 רכיבים נוספים

בסעיף זה נציג מספר רכיבים אחרים שאינם משתייכים לפרקים הקודמים, חלקם חשובים פחות וחלקם חשובים יותר. תת-הסעיפים יסודרו לפי סדר חשיבות הרכיבים.

7.1 מחלקת DBHandler

המחלקה מתקשרת ישירות עם בסיס הנתונים ומממשת פעולות בסיסיות כמו התחברות או ביצוע שאילתה. המחלקה מתפקדת כסוג של Wrapper עבור פעולות בבסיס הנתונים וחוסכת לחלקים בפרוייקט שמתמשים בה התעסקות עם בסיס הנתונים בכלל.

מעבר לפעולות Low-Level כגון connect או executeQuery, כללנו במחלקה פונקציות אשר בתוכן מריצות שאילתות לתפקידים שונים. למשל, פונקציה לקבלת מופע Entity שלטבלת users לפי id של משתמש.

מאחר שמחלקה זו רצה על השרת בלבד, ניתן לבצע את החיבור בעת הצורך פעם אחת ולשמור על מופע המחלקה לאורך זמן. המחלקה אינה בעלת תכונות מופע ואינה קשורה למשתמש, ולכן ניתן לחסוך יצירת מספר מופעים ומספר חיבורים, ולעצבנה בתור Singleton. מבנה זה חוסך עבודה רבה ומייעל באופן משמעותי את הכל הפעולות עם בסיס הנתונים בצד השרת.

סוג מאפיין	שם/חתימה	תיאור
Static Variable	DBHandler instance	משתנה פרטי לשמירת המופע היחיד של המחלקה (Singleton)
Private Constructor	DBHandler()	בנאי פרטי הנקרא רק על ידי getInstance(). אתחול וסימון isConnected=false.
Static Function	DBHandler getInstance()	פונקציית singleton המחזירה את המופע של המחלקה (ואם לא קיים יוצרת כזה באמצעות הבנאי הפרטי).
Function	boolean connect()	מתחברת לבסיס הנתונים, מחזירה true אם החיבור נכשל.
Function	<T> boolean persistEntity(T, Class<T>, Object)	פונקצייה גנרית לעדכון ה-entity בטבלה שלו. אם הוא קיים כבר הוא יעודכן (באמצעות merge) ואם לא הוא יתווסף. פרמטר ה-Object הוא המפתח הראשי (מחלקה או פרימיטיב).
Function	<T> List<T> executeSelectQuery(String, Class<T>, List<Object>, int)	פונקצייה המריצה שאילתה מסוג Select על בסיס הנתונים ומחזירה רשימה של ה-Entities הצפויים להתקבל. הפונקצייה מקבלת רשימה של פרמטרים (עבור prepared statement) ומספר השורות המקסימלי.
Function	int executeAlterQuery(String, List<Object>)	הפונקצייה מריצה שאילתה מסוג Alter (קרי: שאילתה המבצעת עדכון בבסיס הנתונים, למשל הפיכת משתמש ל-admin). הפונקצייה מקבלת רשימת פרמטרים כמו ב-Select ומחזירה את מספר השורות שהושפעו מהרצת השאילתה.

מעבר לפונקציות הרמה הקרובה יותר לבסיס הנתונים שצויינו לעיל, המחלקה מספק פונקציות רבות עיליות יותר, שבהן לא נפרט, כי מדובר בפונקציות פשוטות שלרוב פשוט מריצות שאילתה מסוימת.

7.2 מחלקת Pair

המחלקה אינה בעלת תכולה רבה, אך היא מאוד שימושית ומשומשת במספר מקומות בפרוייקט, בעיקר באלגוריתם ייצור הפגישות על מנת לשמור טווחי זמן.

סוג מאפיין	שם/חתימה	תיאור
------------	----------	-------



משתנה גנרי ראשון לזוג המשתנים.	T1	Generic Class Variable
משתנה גנרי שני לזוג המשתנים.	T2	Generic Class Variable
שומר את האובייקט הראשון.	T1 first	Instance Variable
שומר את האובייקט השני.	T2 second	Instance Variable
בנאי פשוט.	Pair(T1,T2)	Constructor
דורס את toString(). עוטר את ה- toString של first ו- second בסוגריים.	String toString()	Function

7.3 מחלקת PDFHandler

המחלקה דומה באופייה לאופי של מחלקת בסיס הנתונים המתוארת בסעיף 7.1, גם היא מממשת פעולות בסיסיות בעבודה עם השירות. במקרה של מחלקה זאת, השירות אינו בסיס הנתונים אלא יצירת קבצי PDF באמצעות ה-API של ספריית itextpdf. המחלקה מממשת את הפונקציות הקרובות ביותר לספרייה ולבסוף נותנת למשתמש בה את הכלים ליצור קובץ PDF בסיסי, בהתאם לצרכי הפרוייקט כמובן.

סוג מאפיין	שם/חתימה	תיאור
Instance Variable	Document (itextpdf.layout)	משתנה המופע של המסמך שעורכים כרגע.
Instance Variable	final String fileName	שם הקובץ, נקבע דרך הבנאי פעם אחת בלבד.
Instance Variable	final String filePath	מיקום הקובץ, נקבע דרך הבנאי פעם אחת בלבד.
Constructor	PDFHandler (String,String)	מאתחל את שם הקובץ ומיקום הקובץ. אין בנאי דיפולטיבי.
Function	void createDocument()	יצירת קובץ ריק עם הפרטים שהתקבלו בבנאי.
Function	void closeDocument()	סגירת הקובץ ושחרורו למערכת ההפעלה.

גם כאן, בדומה למחלקת DBHandler, ישנן פונקציות עיליות רבות שהן פחות חשובות (אם כי טבען שונה), ניתן דוגמאות לכמה מהן:

סוג מאפיין	שם/חתימה	תיאור
Function	void addTitle(String)	הוסף פסקה בראש הקובץ בעיצוב כותרת.
Function	void addParagraph(String, boolean)	הוסף פסקה בעיצוב סטנדרטי. המשתנה הבוליאני קובע האם לעצב בעיצוב איטלקי.
Function	void addFooter(String)	הוסף Footer בתחתית כל עמוד בקובץ עם הסגנון הקבוע מראש והטקסט הנתון.

7.4 מחלקת AppointmentExporter

המחלקה מספקת פונקצייה סטטית עיקרית המגדירה קבועים ומשתמשת במחלקת PDFHandler על מנת ליצור קובץ מפגישה נתונה.

סוג מאפיין	שם/חתימה	תיאור
Static Constant	final String ...PATH	מיקום של התיקיה לשמירת הקובץ (קבוע בשרת).
Static Constant	final String FOOTER	הטקסט שיופיע ב- footer.



“Hospital Appointment) המערכת את המתאר את קבוע המערכת (System”, מופיע בראש הקובץ.	final String PUBLISHER	Static Constant
“Hospital) למעלה מופיע המסמך, (Appointment”	final String TITLE	Static Constant

7.5 מחלקת Utils

המחלקה כוללת פונקציות עזר כלליות יותר שלא השתייכו למחלקות אחרות מידי. אין צורך לתאר אותן אחת אחת. המחלקה כוללת פונקציות סטטיות בלבד והמופע שלה חסר משמעות (לו הייתה ב-Java אופציה למחלקה סטטית או תכנות פונקציונלי זה היה מקום טוב להשתמש בה).

7.6 מחלקת BirthDateValidator

כדי להדגים את השימוש ב-Validator שאנו יוצרים בעצמנו, המחלקה מממשת את ממשק Validator<> של JSF ומשמשת לוידוא התאריך המתקבל מהמשתמש בעת ההרשמה. ה- validator בודק שהתאריך שניתן לא ממקם את המשתמש מעל גיל קבוע של 120 ושהוא לא בעל גיל שלילי. את רכיב זה ניתן לממש בקלות בעזרת אלמנטים קיימים של PrimeFaces 12.0.0, ואנו אכן עושים זאת בחלקים אחרים בפרוייקט.

סוג מאפיין	שם/חתימה	תיאור
Class Annotation	@FacesValidator (“birthDateValidator”)	הצהרה על המחלקה בתור validator
Interface Implementation	Validator<Date>	מימוש ממשק ה- validator של JSF כדי לממש את הפונקציות הדרושות.
Function	void validate (FacesContext, UIComponent, Date) throws ValidatorException	מימוש פונקציית הממשק validate, וידוא התאריך קלט וזריקת ValidatorException במידה ואינו עומד במגבלות.
Static Constant	final int MAX_AGE	הגיל המקסימלי של המשתמש (120).

7.7 מחלקת DateConverter

כדי להדגים את השימוש ב-Converter שאנו יוצרים בעצמנו, המחלקה מממשת את ממשק Converter<> של JSF ומשמשת להמרת תאריך ממחרוזת בפורמט שלבחרתנו לאובייקט Date. גם את רכיב זה ניתן לממש בקלות בעזרת אלמנטים קיימים של PrimeFaces 12.0.0.

סוג מאפיין	שם/חתימה	תיאור
Class Annotation	@FacesConverter (“dateConverter”)	הצהרה על המחלקה בתור converter
Interface Implementation	Converter<Date>	מימוש ממשק ה- converter של JSF כדי לממש את הפונקציות הדרושות.
Function	Date getAsObject (FacesContext, UIComponent, String)	מימוש פונקציית הממשק, המרת ערך המחרוזת לאובייקט Date באמצעות אובייקט SimpleDateFormat.
Function	String getAsString (FacesContext, UIComponent, Date)	מימוש פונקציית הממשק, המרת ערך אובייקט למחרוזת להצגה בשדה JSF באמצעות אובייקט SimpleDateFormat.

