

מטלה 3- מסמך ניתוח

מגשים:

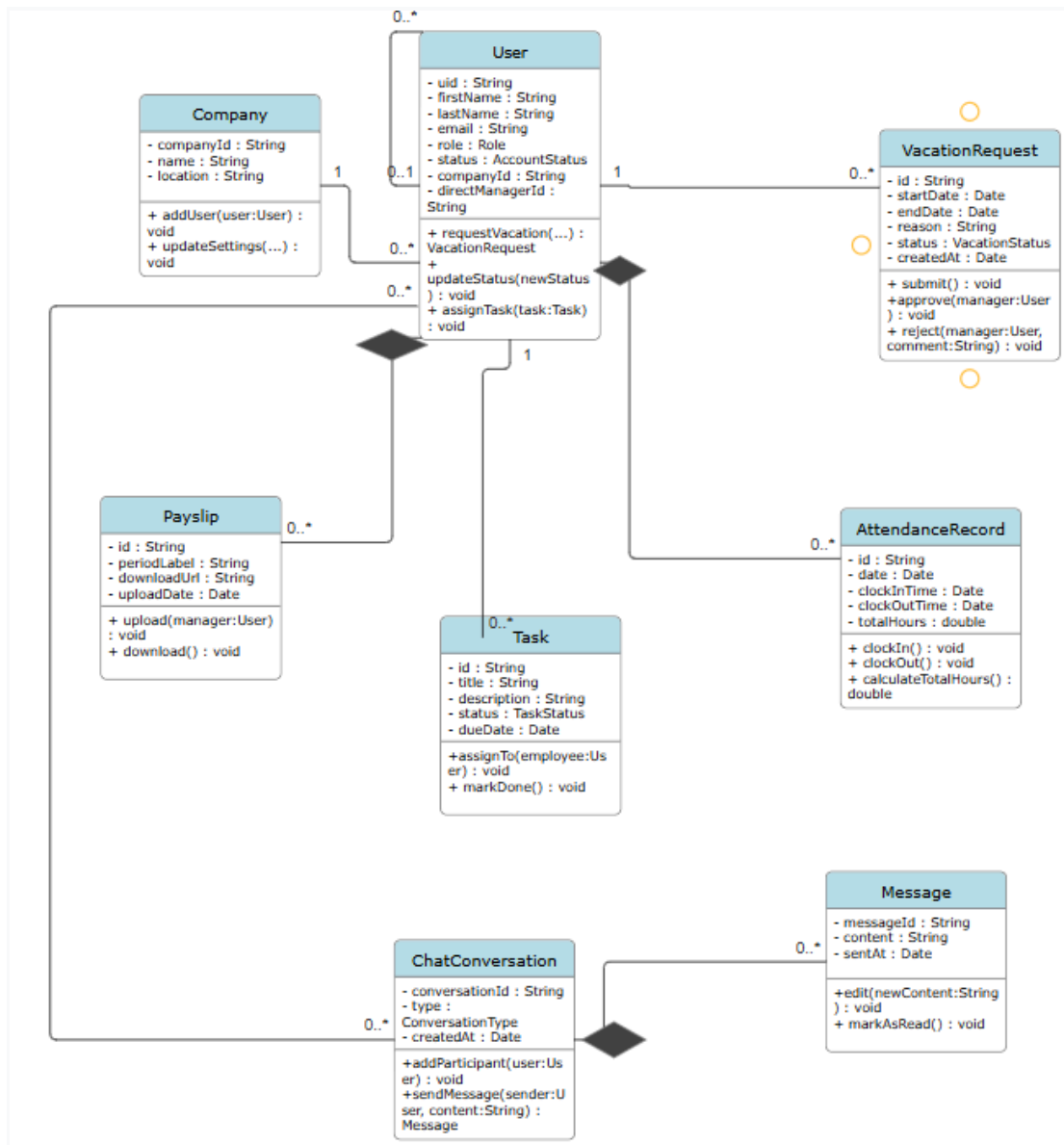
רפאל קואפיק 337614747

שי קרונפלד 322234782

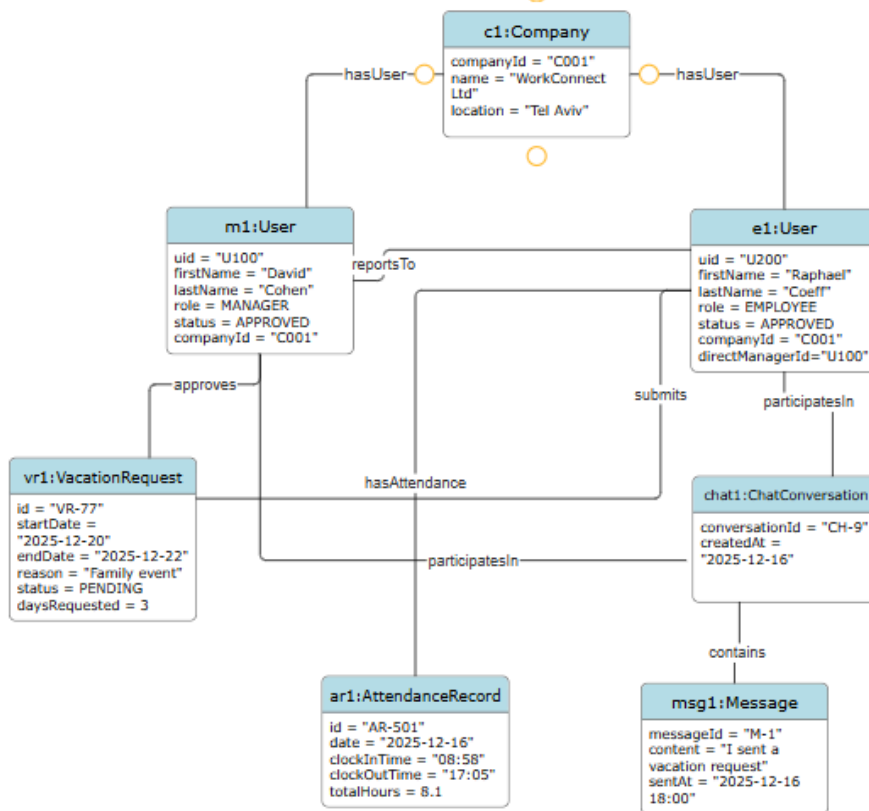
משה אסקרוב 314085986

חלק א'- דיאגרמות

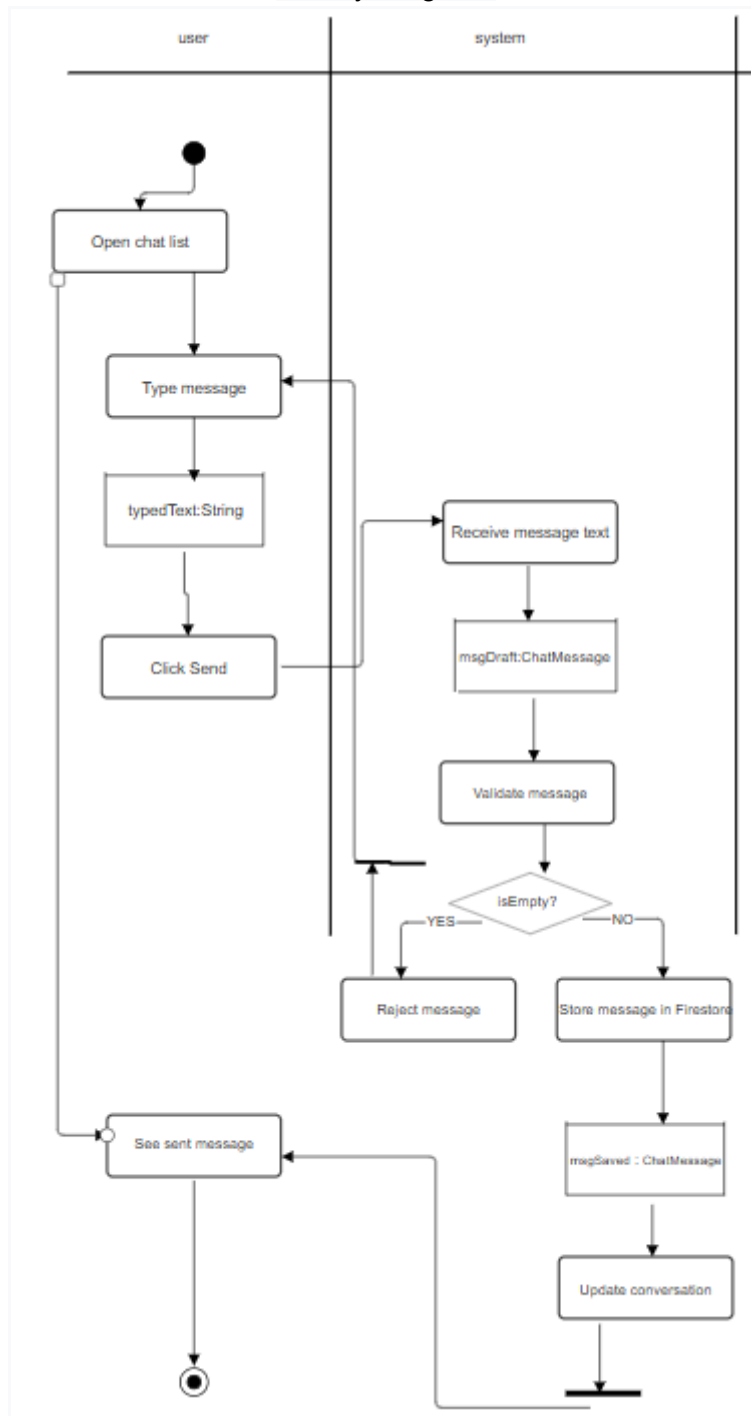
Class Diagram:



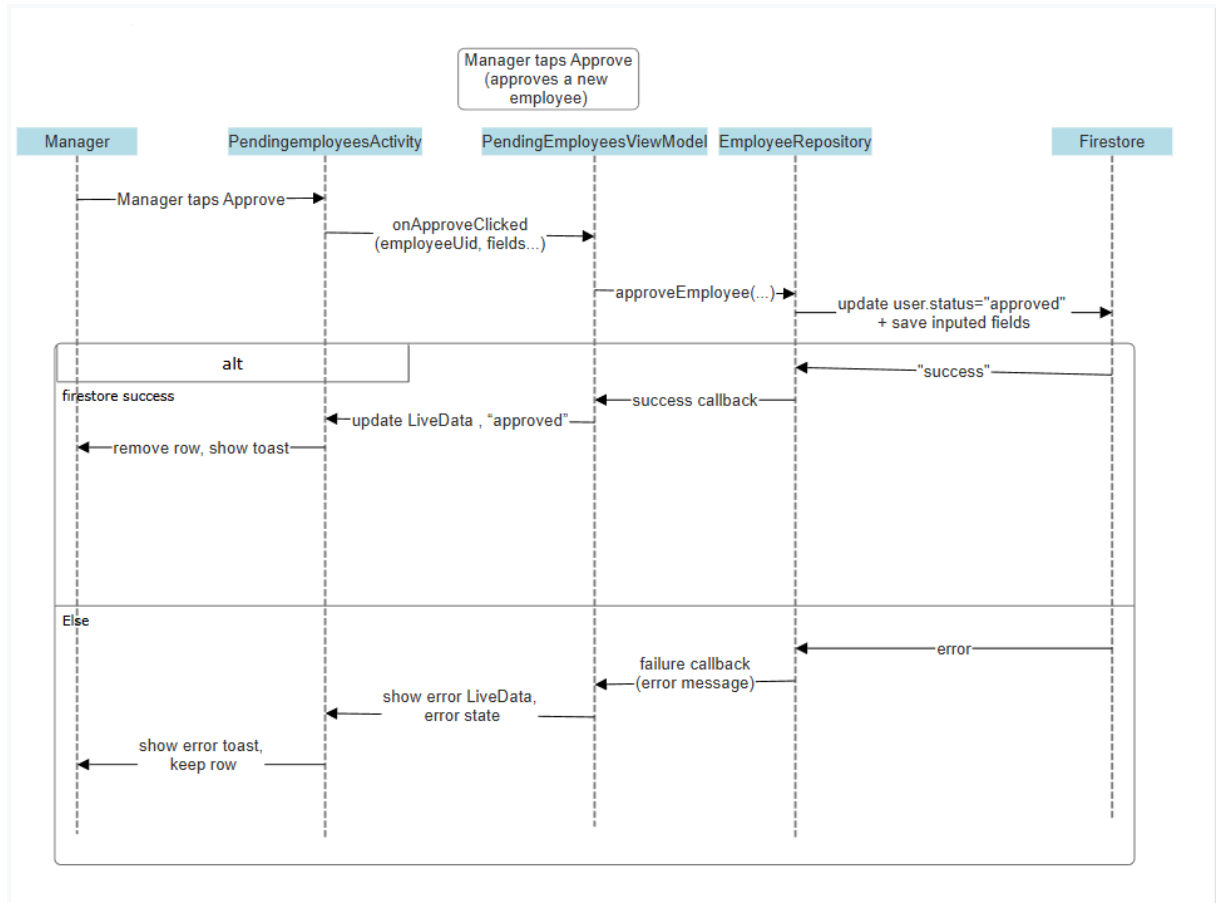
Object Diagram:



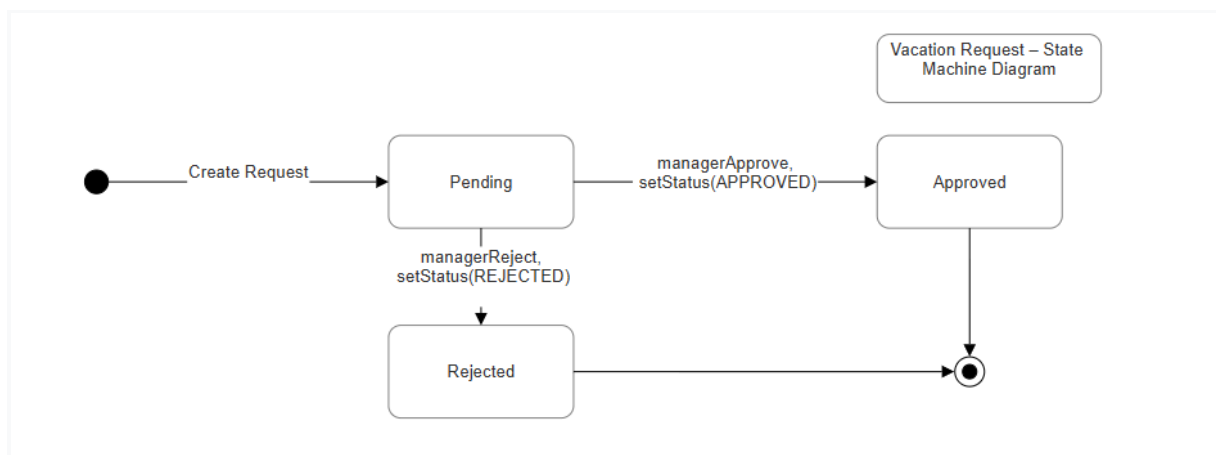
Activity Diagram:



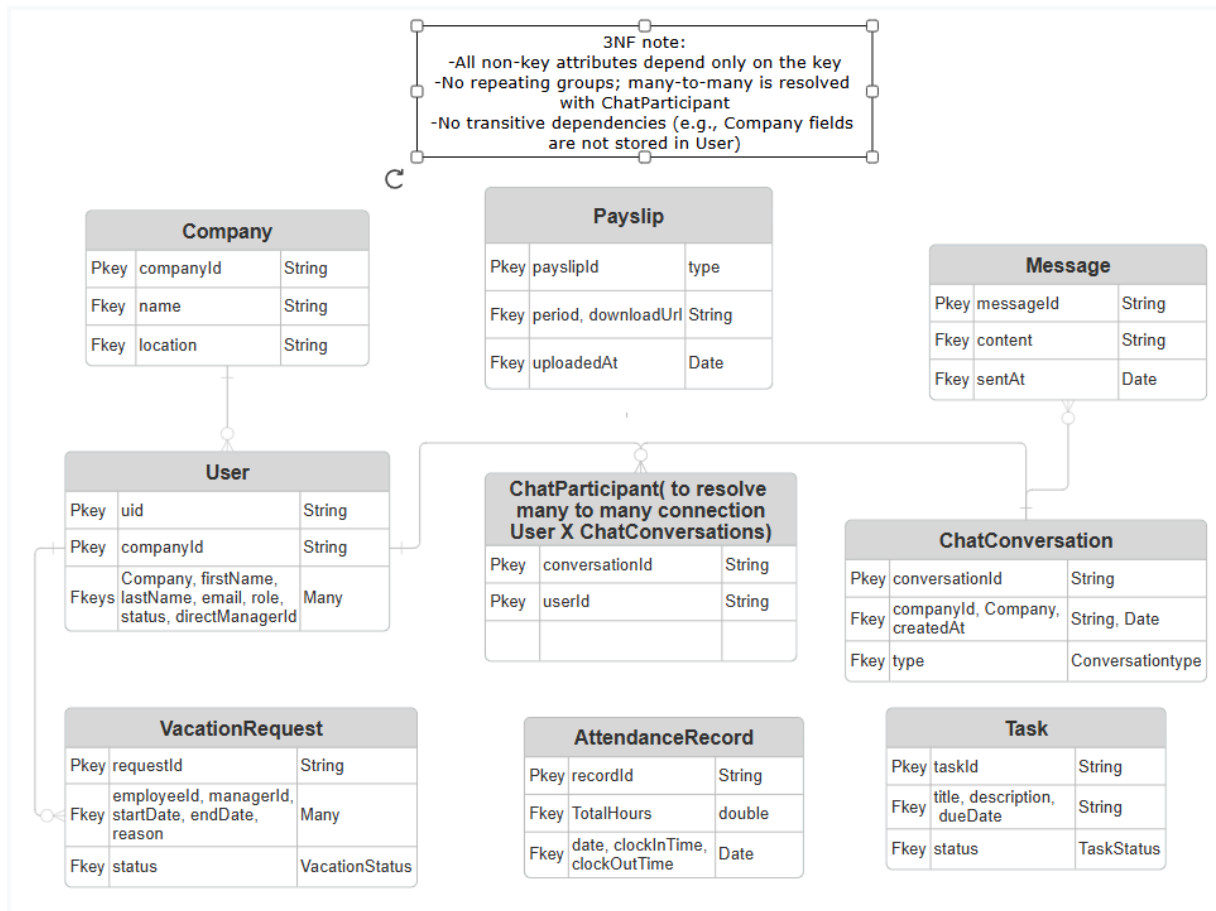
Sequence Diagram:



State Machine Diagram:



ERD (Entity-Relationship Diagram):



חלק ב' - מבנה מסכי המערכת

רצינו להבהיר לפני פירוט חלק זה מספר נקודות:

- יש במערכת מסכים שמשמשים גם עובדים וגם מנהלים. הסיבה לכך היא שמנהל הוא גם עובד, ולכן אופן השימוש והזרימה במסכים האלה זהים. כדי להימנע מחזרות, תיארנו את המסכים והזרימה המשותפים רק בחלק של העובדים. בחלק של המנהלים פירטנו רק את המסכים והפעולות שייחודיים למנהל.
- בפרק הזרימה תיארנו בצורה מפורטת תהליכים שכבר התחלנו לממש בפועל באפליקציה. לעומת זאת, תהליכים שעדיין לא מומשו תוארו באופן כללי יותר, כדי להסביר את הרעיון והכיוון שאליו אנחנו מתכננים להגיע בהמשך הפיתוח.

1. רשימת מסכים מרכזיים:

- מסכים להרשמה (גם לעובדים וגם למנהלים) -
 - Login (כניסה)
 - Register Type (בחירת סוג הרשמה)
 - Register Company (רישום חברה חדשה – מנהל מצרף חברה למערכת)
 - Register Employee (רישום עובד חדש)
- מסכים עבור כלל העובדים -
 - Employee Home Activity (מסך בית לעובד)
 - my Vacations Activity (רשימת החופשות של העובד)
 - מסך New Vacation Request Activity (בקשת חופשה חדשה)
 - my Shifts Activity (רשימת המשמרות של העובד)
 - מסך Add Availability To Shifts Activity (הגשת זמינות משמרות)
 - מסך My Salary Slips Activity (תלושי שכר שלי)
 - מסך chat (צ'אט)
 - מסך video call (שיחת וידאו)
- מסכים נוספים עבור המנהלים -
 - Manager Home Activity (מסך בית למנהל)
 - Pending Employees Activity (עובדים שממתינים לאישור הרשמה)
 - מסך Vacation Requests Management Activity (ניהול בקשות חופשה)
 - מסך Company Settings Activity (מציג את הגדרות החברה)
 - מסך Shifts Management Activity (ניהול משמרות)
 - מסך Salary Slips Upload Activity (העלאת תלושי שכר)

2. קשרים וזרימת ניווט בין מסכים:

• זרימה ראשית - התחברות והרשמה:

- האפליקציה נפתחת - LoginActivity:
 - אם המשתמש כבר מחובר, יכניס מייל וסיסמא וילחץ על Login.
 - נבדוק את ה- role מה- Firestore:
 - אם manager - אז LoginActivity → ManagerHomeActivity
 - אם employee - אז LoginActivity → EmployeeHomeActivity
 - אם ילחץ על כפתור Register אז LoginActivity → RegisterTypeActivity
- RegisterTypeActivity:
 - אם ילחץ על כפתור Create new company אז RegisterTypeActivity → RegisterCompanyActivity
 - אם ילחץ על כפתור Join existing company אז RegisterTypeActivity → RegisterEmployeeActivity
 - אם ילחץ על כפתור Back יחזור חזרה למסך הקודם. כלומר: RegisterTypeActivity → LoginActivity
- RegisterCompanyActivity:
 - מילוי פרטי חברה + מנהל.
 - אם ילחץ על כפתור Create company:
 - יצירת משתמש מנהל + מסמך חברה וגם RegisterCompanyActivity → ManagerHomeActivity
 - אם ילחץ על כפתור Back יחזור חזרה למסך הקודם. כלומר: RegisterCompanyActivity → RegisterTypeActivity
- RegisterEmployeeActivity:
 - מילוי שם מלא, אימייל, סיסמה, קוד חברה.
 - אם ילחץ על כפתור Create employee account:
 - יצירת משתמש עובד עם status = pending
 - Toast: הרשמה ממתינה לאישור מנהל.
 - וכן, RegisterEmployeeActivity → LoginActivity כי התהליך הסתיים כרגע.
 - אם ילחץ על כפתור Back יחזור חזרה למסך הקודם. כלומר: RegisterEmployeeActivity → RegisterTypeActivity

• זרימה בצד העובד

- אם הגיע ל- EmployeeHomeActivity (אחרי Login כאדם עם role employee)
 - כפתורים בניווט:
 - Attendance → myAttendanceActivity
 - My shifts → MyShiftsActivity
 - My vacations → VacationRequestsActivity
 - My salary slips → MySalarySlipsActivity
 - chats → my chats Activity
 - video calls → my video calls activity
 - My profile → UserProfileActivity

- חזרה למסך ההתחברות (LoginActivity) → Logout.

○ אם הגיע ל- myAttendanceActivity

- העובד רוצה לסמן שהגיע לעבודה/ יצא ממנה או שרוצה לראות את היסטוריית הנוכחות שלו.
- עם פתיחת המסך מוצג לוח שנה בפורמט חודשי כאשר היום הנוכחי יהיה בולט ויוצגו נתוני כניסה ויציאה שלו מאותו יום.
- אם ילחץ על entry / exit תדווח כניסה/ יציאה בהתאם ותוצג לו התראה על השעה שנרשמה.
- כפתור כניסה או יציאה יקבע בהתאם ללחיצה הקודמת שנלחצה בנוכחות.
- כפתור Back יחזיר חזרה למסך הקודם. כלומר:
myAttendanceActivity → EmployeeHomeActivity

○ אם הגיע ל- MyShiftsActivity

- העובד רוצה לראות את המשמרות שלו.
- עם פתיחת המסך מוצג לוח שנה חודשי, כאשר היום הנוכחי מודגש באופן ברור.
- כברירת מחדל יוצגו המשמרות אליהן העובד משובץ ביום הנוכחי.
- אם ילחץ על יום אחר בלוח השנה, יוצגו לעובד המשמרות אליהן הוא משובץ בתאריך שנבחר.
- אם ילחץ על set availability for this month אז העובד רוצה להזין זמינות חדשה ולכן: MyShiftsActivity → AddAvailabilityToShiftsActivity.
- כפתור Back יחזיר חזרה למסך הקודם. כלומר:
MyShiftsActivity → EmployeeHomeActivity

○ אם הגיע ל- AddAvailabilityActivity

- העובד רוצה להזין זמינות חדשה למשמרות.
- אם ילחץ על ASSIGN / UPDATE AVAILABILITY יפתח חלון קטן בו המשתמש יבחר את המשמרת אותה הוא רוצה לסמן. ברגע שיבחר משמרת יפתח חלון נוסף בו המשתמש יבחר את ההעדפה למשמרת (CAN / PREFER NOT / CAN'T).
- בחלוניות יש כפתור cancel שאם המשתמש ילחץ עליו החלונית תיסגר ללא שמירה.
- אם ילחץ על DONE השינויים ישמרו.

○ אם הגיע ל- MyVacationsActivity

- יוצגו רשימת בקשות החופשה של המשתמש.
- אם ילחץ על כפתור New Vacation, העובד רוצה לפתוח בקשה לחופשה חדשה ולכן, MyVacationsActivity → NewVacationRequestActivity.
- כפתור Back יחזיר חזרה למסך הקודם. כלומר:
MyVacationsActivity → EmployeeHomeActivity

○ אם הגיע ל- NewVacationRequestActivity

- העובד יבחר תאריכים לחופשה ויכתוב סיבה.
- כפתור Send request:
- יוצר בקשה חדשה (VacationRequest) עם סטטוס pending.
- Toast הצלחה.
- חזרה ל- EmployeeHomeActivity (*) → finish()

- כפתור Back יחזיר חזרה למסך הקודם. כלומר:
NewVacationRequestActivity → MyVacationsActivity

○ אם הגיע ל-MyTasksActivity

- העובד רוצה לראות את המשימות שהוקצו לו (משימות פתוחות/בטיפול/הושלמו).
- עם פתיחת המסך תוצג רשימת משימות של העובד, כאשר לכל משימה יוצגו פרטים בסיסיים כגון: כותרת, תיאור קצר, תאריך יעד וסטטוס.
- אם העובד ילחץ על משימה מסוימת, יוצגו לו פרטי המשימה המלאים (כגון תיאור מלא, תאריך יעד, מי הקצה את המשימה והסטטוס).
- אם ילחץ על כפתור עדכון סטטוס העובד יוכל לשנות את הסטטוס והמערכת תשמור את העדכון ותציג הודעת הצלחה.
- כפתור Back יחזיר חזרה למסך הקודם. כלומר:
MyTasksActivity → EmployeeHomeActivity

○ אם הגיע ל-ChatsActivity

- העובד רוצה לנהל שיחות עם משתמשים אחרים בחברה.
- עם פתיחת המסך תוצג רשימת צ'אטים קיימים של העובד (לפי שיחות אחרונות).
- אם העובד ילחץ על שיחה מסוימת, תיפתח תצוגת השיחה ובה יוצגו ההודעות הקודמות.
- העובד יוכל להקליד הודעה וללחוץ Send, ההודעה תישלח ותופיע מיד בשיחה.
- אם ילחץ על יצירת שיחה חדשה (New Chat), יוכל לבחור משתמש מהרשימה ולהתחיל שיחה חדשה.
- כפתור Back יחזיר חזרה למסך הקודם. כלומר:
ChatsActivity → EmployeeHomeActivity

○ אם הגיע ל-VideoCallActivity

- העובד רוצה לבצע שיחת וידאו עם משתמש אחר בחברה.
- עם פתיחת המסך תוצג אפשרות לבחור איש קשר.
- אם העובד יבחר משתמש וילחץ על Start Call, תתחיל שיחת וידאו ותוצג תצוגת מצלמה ומיקרופון.
- במהלך השיחה העובד יוכל לבצע פעולות בסיסיות כגון: השתקת מיקרופון, כיבוי מצלמה, וסיום שיחה.
- בסיום שיחה (End Call) המשתמש יחזור למסך הקודם.
- כפתור Back יחזיר חזרה למסך הקודם. כלומר:
VideoCallActivity → EmployeeHomeActivity

○ אם הגיע ל-MyProfileActivity

- העובד רוצה לצפות בפרטים האישיים שלו.
- עם פתיחת המסך יוצגו פרטי העובד כמו: שם מלא, אימייל, תפקיד, שם החברה וכו' וכן יוצג אזור ייעודי לתלושי השכר.
- בחלק של תלושי השכר תוצג רשימה לפי חודשים.
- אם העובד ילחץ על תלוש שכר מסוים תיפתח אופציה של הצגה או הורדה של התלוש.
- כפתור Back יחזיר חזרה למסך הקודם. כלומר:
MyProfileActivity → EmployeeHomeActivity

- זרימה בצד המנהל

- אם הגיע ל- ManagerHomeActivity (אחרי Login כאדם עם role manager)

- כפתורים בניווט:

- אישי:

- . My vacations → VacationRequestsActivity
- .My shifts → MyShiftsActivity
- .My salary slips → MySalarySlipsActivity
- my tasks → MyTasksActivity
- my chats → my chats Activity
- my video calls → my video calls activity
- .all employees → employees list

- ניהולי:

- .Approve new users → PendingEmployeesActivity
- .Vacation requests → PendingVacationRequestsActivity
- .manage shifts → ManageShiftsActivity
- .upload salary slips → UploadSalarySlipsActivity
- .company setting → CompanySettingActivity

- חזרה למסך ההתחברות (LoginActivity) → Logout.

- אם הגיע ל- PendingEmployeesActivity

- מוצג RecyclerView של עובדים חדשים שממתינים לאישור:

- אם ילחץ על Approve לעובד אז נפתח דיאלוג אישור עם Role. ו-
- PendingEmployeesActivity → dialog_approve_employee
- אם ילחץ על על Reject אז הסטטוס ישתנה ל- rejected.
- אם ילחץ על כפתור Back יחזור חזרה למסך הקודם. כלומר:
- PendingEmployeesActivity → ManagerHomeActivity

- אם הגיע ל- dialogApproveEmployee

- המנהל צריך למלא את הפרטים הבאים עבור העובד: מנהל ישיר, מספר ימי החופשה בחודש שהעובד זכאי, שם המחלקה אליה הוא נכנס, באיזה קבוצה ומה השם של התפקיד שלו.
- הנתונים נשמרים ב- Firestore והסטטוס משתנה ל- approved.
- הרשימה מתעדכנת אוטומטית דרך listener.
- אם ילחץ על כפתור Back יחזור חזרה למסך הקודם. כלומר:
- dialogApproveEmployee → PendingEmployeesActivity

- אם הגיע ל- PendingVacationRequestsActivity

- מוצג RecyclerView של רשימת בקשות לחופשות של העובדים שממתינים לאישור.
- מתוך רשימת הבקשות יוכל ללחוץ על:
- כפתור Approve / Reject לכל בקשה.
- אם ילחץ על כפתור Back יחזור חזרה למסך הקודם. כלומר:
- PendingVacationRequestsActivity → ManagerHomeActivity

- אם הגיע ל- ManageAttendanceActivity

- המנהל רוצה לצפות ולנהל את נתוני הנוכחות של העובדים.
- עם פתיחת המסך תוצג רשימת העובדים שלו.

- המנהל יוכל לבחור עובד מהרשימה, ולאחר הבחירה יוצגו נתוני הנוכחות של העובד הנבחר בפורמט של רשימה לפי תאריכים עם סטטוס נוכחות.
- המנהל יוכל לבחור יום מסוים מתוך רשימת הנוכחות של העובד כדי לצפות בפרטים מלאים של אותו יום.
- המנהל יוכל לעדכן נתוני כניסה ויציאה של עובדים וכך לאשר דיווחי נוכחות.
- בסיום המנהל יצטרך ללחוץ על כפתור SAVE לשמירה.
- כפתור Back יחזיר חזרה למסך הקודם. כלומר:
ManageAttendanceActivity → ManagerHomeActivity

■ אם הגיע ל- ManageShiftsActivity

- המנהל ירצה לשבץ משמרות לעובדים שלו.
- בעת פתיחת המסך יוצגו למנהל הקבוצות אותן הוא צריך לשבץ. בעת לחיצה על קבוצה מסוימת יפתח לוח שנה בתצוגה חודשית.
- בעת לחיצה על תאריך מסוים המשתמש ילחץ על כפתור ASSIGN ויוצגו לו המשמרות של אותו יום.
- אם ילחץ על משמרת מסוימת, תוצג לו רשימה שכתובים בה העובדים שהגישו זמינות לאותה משמרת.
- המנהל יוכל ללחוץ על עובד מסוים אותו הוא רוצה לשבץ.
 - תיפתח חלונית שתציג את פרטי העובד.
 - אם ילחץ על ASSIGN ולשבץ את העובד.
 - אם ילחץ על CANCEL החלונית תיסגר ללא שיבוץ.
- אם ילחץ על כפתור Back יחזור חזרה למסך הקודם. כלומר:
ManageShiftsActivity → ManagerHomeActivity

■ אם הגיע ל- UploadSalarySlipsActivity

- המנהל יעלה תלושי שכר לעובדים שלו.
- אם ילחץ על כפתור Back יחזור חזרה למסך הקודם. כלומר:
UploadSalarySlipsActivity → ManagerHomeActivity

■ אם הגיע ל- CompanySettingsActivity

- המנהל יעדכן ערכים על החברה כמו: מיקום, מינוס חופשה מותר, ברירת מחדל לצבירה.
- כפתור Save – שמירה ב- Firestore.
- אם ילחץ על כפתור Back יחזור חזרה למסך הקודם. כלומר:
CompanySettingsActivity → ManagerHomeActivity

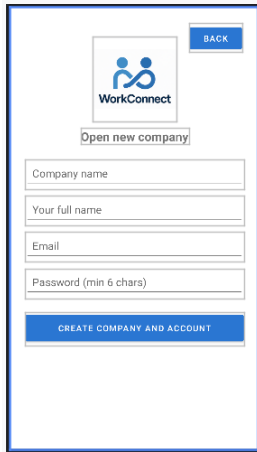
3. ייצוג חזותי:

מסכי התחברות והרשמה-

מסך Register Company

מסך Register Type

מסך Login



WorkConnect

Open new company

Company name

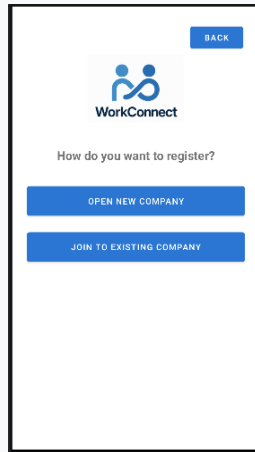
Your full name

Email

Password (min 6 chars)

CREATE COMPANY AND ACCOUNT

BACK



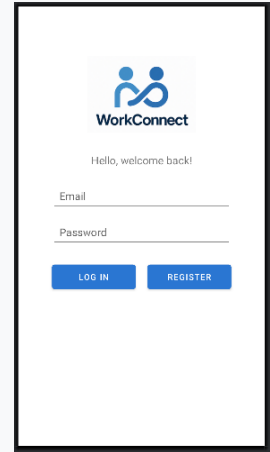
WorkConnect

How do you want to register?

OPEN NEW COMPANY

JOIN TO EXISTING COMPANY

BACK



WorkConnect

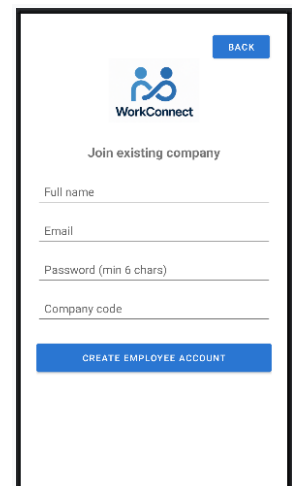
Hello, welcome back!

Email

Password

LOG IN REGISTER

מסך Register Employee



WorkConnect

Join existing company

Full name

Email

Password (min 6 chars)

Company code

CREATE EMPLOYEE ACCOUNT

BACK

מסך New Vacation Request:

A form titled "New Vacation Request" with a "Back" button at the top right. It contains three input fields: "Start date", "End date", and "Reason for vacation". Below these fields is a blue "Send request" button.

מסך my Vacations:

A screen titled "my Vacations" with a "Back" button at the top right. It features a table with two columns: "Vacation Date" and "Status". The table contains ten rows, each labeled "Item 0" through "Item 9". At the bottom of the screen is a blue "New Request" button.

מסך Employee Home:

An "Employee Home" screen with a "Logout" button at the top right. It displays the user's name "Hello, Employee" and company "Company: My Company". Below this is a section "Employee actions" with a grid of buttons: "Attendance", "My shifts", "My vacations", "My tasks", "Chats", and "Video calls". At the bottom is a blue "My profile" button.

מסך all chats:

An "all chats" screen with a search bar at the top labeled "Search employee...". Below the search bar, it lists a chat with "Raphael Coeffic" and a timestamp "Dec 16, 2023 6:20:09 PM". Below this is a section labeled "chat 0" with a timestamp "Dec 16, 2023 2:07:49 PM".

מסך chat:

A "chat" screen showing a conversation. It starts with a grey bubble saying "How are you?" at 6:20 PM. This is followed by a green bubble saying "Hi Raphael I am fine and you?" at 6:28 PM. At the bottom is a text input field labeled "Écrire un message..." and a send button.

מסך Pending vacation request:

A "Pending vacation request" screen with a "Back" button at the top right. It displays a list of "Vacation requests (pending)" with items labeled "Item 0" through "Item 5".

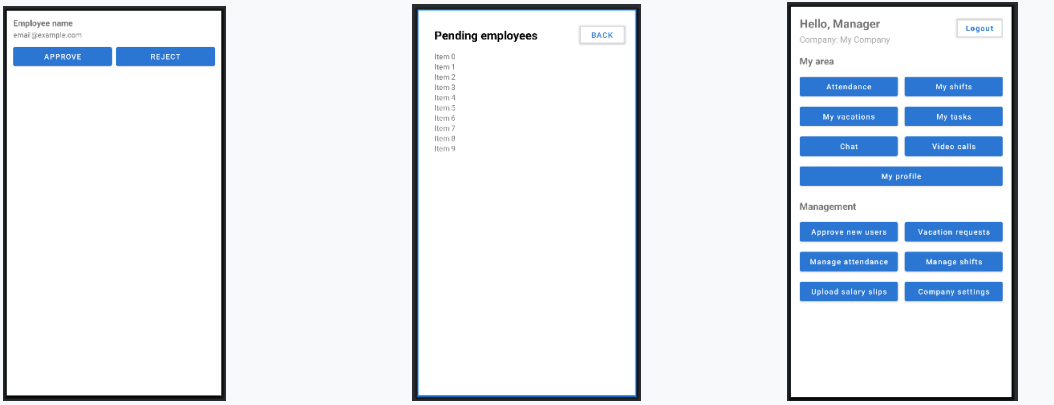
מסך add availability:

An "add availability" screen titled "Set Availability" for "January 1970". It shows a calendar with days 1 through 31. Day 1 is highlighted with a green circle. Below the calendar is a section "Tap a day" with the text "Tap a day to view what you already set." and a button "ASSIGN / UPDATE AVAILABILITY". At the bottom is a blue "DONE" button.

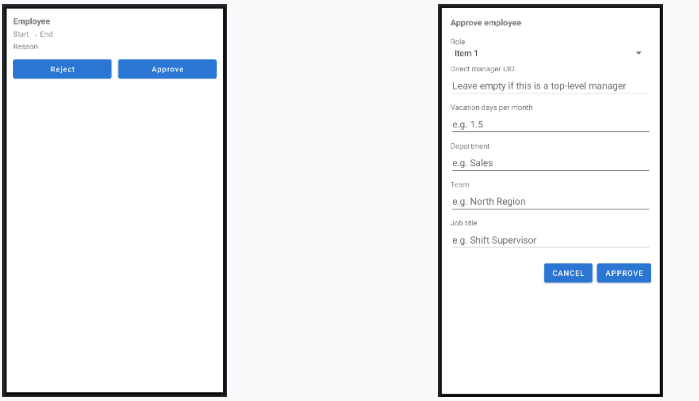
מסך my shifts:

A "my shifts" screen titled "My Shifts" for "January 1970". It shows a calendar with days 1 through 31. Day 1 is highlighted with a green circle. Below the calendar is a legend with three items: "Assigned" (green circle), "Prefer not" (yellow circle), and "Unavailable" (red circle). Below the legend is a section "Select a day" with the text "Tap a day to see your scheduled shifts." and a blue button "SET AVAILABILITY FOR THIS MONTH".

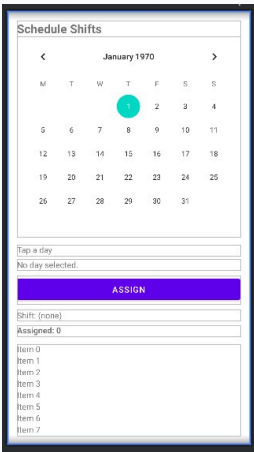
מסך Manager Home מסך Pending Employees מסך item pending employee



מסך dialog approve employee מסך item pending vacation request



מסך manage shifts



חלק ג' - מבנה הקוד ותבניות עיצוב עבור Android Studio

1. תיקיות הפרויקט שלנו הן:

• :ui

- בתיקייה זו נמצאים Activities/Fragments של המסכים.
- תפקיד הקבצים בתיקייה זו: להציג מידע, לקבל קלט מהמשתמש, לנווט בין מסכים, ולהאזין ל-LiveData מה-ViewModel.
- הקבצים בתיקייה זו נמצאים בתוך יש 3 תתי תיקיות בשביל סדר:
 - תיקיית auth : בה יש הקבצים שאחראיים על רישום והתחברות.
 - תיקיית manager : בה יש את הקבצים שקשורים למנהל.
 - תיקיית employee : בה יש את הקבצים שקשורים לעובד.

• :viewmodel

- בתיקייה זו נמצאים ViewModels.
- תפקיד הקבצים בתיקייה זו: לשמור מצב מסך, לבצע לוגיקה שקשורה למסך, ולדבר עם ה-Repository.
- הקבצים בתיקייה זו נמצאים בתוך יש 3 תתי תיקיות בשביל סדר:
 - תיקיית auth : בה יש הקבצים שאחראיים על רישום והתחברות.
 - תיקיית manager : בה יש את הקבצים שקשורים למנהל.
 - תיקיית employee : בה יש את הקבצים שקשורים לעובד.

• :repository

- "מקור אמת". המסכים וה-ViewModel לא יודעים מאיפה הנתונים מגיעים, רק קוראים לפונקציות של ה-Repository.

• :models

- בתיקייה זו יש אובייקטים שמייצגים ישויות במערכת: User, Employee, Manager, Company, VacationRequest.

• :adapters

- מתאמים של RecyclerView: למשל, VacationRequestsAdapter, PendingEmployeesAdapter.
- תפקיד הקבצים בתיקייה זו: לקשר בין רשימת נתונים לבין תצוגת שורות.

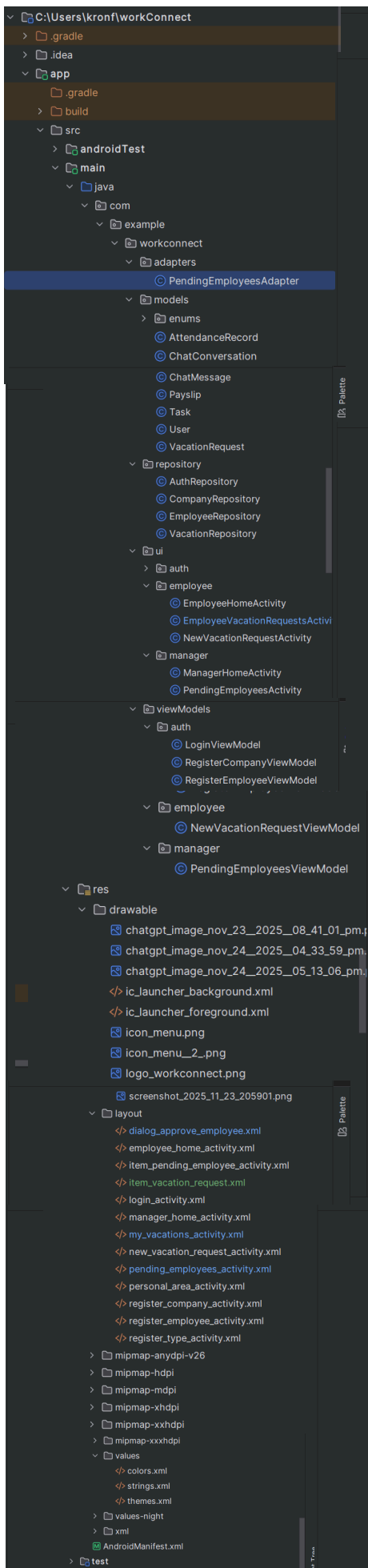
בחרנו דווקא במבנה הזה כי הוא מתאים הכי טוב לדרישות של הפרויקט שלנו ולסוג הבעיות שהוא פותר. המבנה נותן סדר ברור ומונע "בלאגן" בקוד כשמוסיפים עוד מסכים ופיצ'רים.

• הפרדת אחריות ברורה:

- ui - אחראי רק על תצוגה, ניווט וקלט מהמשתמש.
- viewmodel - מנהל את מצב המסך ואת הלוגיקה שקשורה למסך (למשל: טעינה, הודעות שגיאה, מתי להציג רשימה).
- models - מגדיר את מבני הנתונים (Company, User, VacationRequest וכו').
- זה חשוב כי ככה לא "מערבבים" קוד תצוגה עם קוד נתונים, והכול נהיה קריא וקל לתחזוקה.

- התאמה טבעית ל-MVVM (הארכיטקטורה שבחרנו):
 - MVVM עובד הכי טוב כשיש שכבות ברורות:
 - UI → ViewModel → Repository → Firebase
 - המבנה הזה ממש "מכריח" לעבוד נכון לפי MVVM ומונע מצב שבו Activity מבצע קריאות ישירות ל-Firebase (שזה מקור לבאגים וקוד מסורבל).
- יכולת גדילה של הפרויקט בלי לשבור את הקיים:
 - בפרויקט שלנו אנחנו צפויים להוסיף עוד מסכים ותהליכים (לדוגמה: הגדרות חברה, מדיניות חופשות, דוחות מנהל, חוקים כמו מינוס ימי חופשה). וככה כל פיצ'ר חדש נכנס למקום קבוע:
 - UI למסכים
 - ViewModel ללוגיקה
 - Repository לנתונים
 - אפשר להוסיף בלי "למרוח" קוד בכל הפרויקט.
- עבודה נכונה עם Firebase ושינויים עתידיים:
 - Firebase הוא מקור נתונים חיצוני, ויכול לקרות שיום אחד נרצה להחליף חלק מהקריאות או לשנות מבנה מסמכים ב-Firebase. אם כל הגישה לנתונים מרוכזת ב-Repository, אנחנו נשנה במקום אחד במקום לעבור על כל המסכים.
- קל יותר לבדוק ולתקן תקלות:
 - למשל כשיש בעיה בבקשות חופשה:
 - אם זה בתצוגה אז מחפשים ב-ui.
 - אם זה בנתונים או בשאילתה אז מחפשים ב-repository.
 - אם זה במצב מסך (loading/error) אז מחפשים ב-viewmodel
 - וזה מקצר משמעותית זמן דיבוג.
- עקביות בין מודולים שונים:
 - המערכת שלנו מחולקת תפקידים. לכן היה לנו חשוב שגם ה-UI למשל יהיה מסודר לפי תחומים. זה עוזר להבין מהר איפה כל דבר נמצא ומונע כפילויות.

צירוף צילום מסך של מבנה התיקיות בקוד:



2. בחרנו בתבנית עיצוב MVVM נפרט את היתרונות של בחירה דווקא בתבנית זו:

- תחזוקתיות:

- שינוי בלוגיקה של תהליך לא דורש שינוי במסכים עצמם.
- הקוד מחולק לשכבות: $UI \leftrightarrow ViewModel \leftrightarrow Repository \leftrightarrow Firebase$ ולכן קל יותר למצוא קבצים ולהבין "איפה כל דבר נמצא".

- סקיילביליות:

- קל להוסיף פיצ'רים חדשים (למשל: סוגי חופשות, דוחות למנהל, התראות) בלי "לשבור" קוד קיים.
- אפשר להוסיף מקור נתונים נוסף בעתיד בלי לשנות את המסכים - רק לעדכן את ה-Repository.

- ביצועים וחווית משתמש:

- ViewModel שומר מצב ולא "מאבד מידע" בסיבובי מסך/שחזור Activity, ולכן פחות קריאות חוזרות ל-Firebase.
- אפשר להוסיף אופטימיזציה בשכבת ה-Repository בלי לגעת ב-UI.
- RecyclerView + Adapter מאפשרים הצגה יעילה של רשימות (כמו בקשות חופשה/עובדים בהמתנה).

- בדיקות ואיכות קוד:

- קל יותר לבדוק לוגיקה ב-ViewModel וב-Repository (יחידות בדיקה), כי היא לא תלויה ישירות במסך או ב-Android UI.
- הפרדת אחריות מורידה באגים ומקלה על דיבוג.