

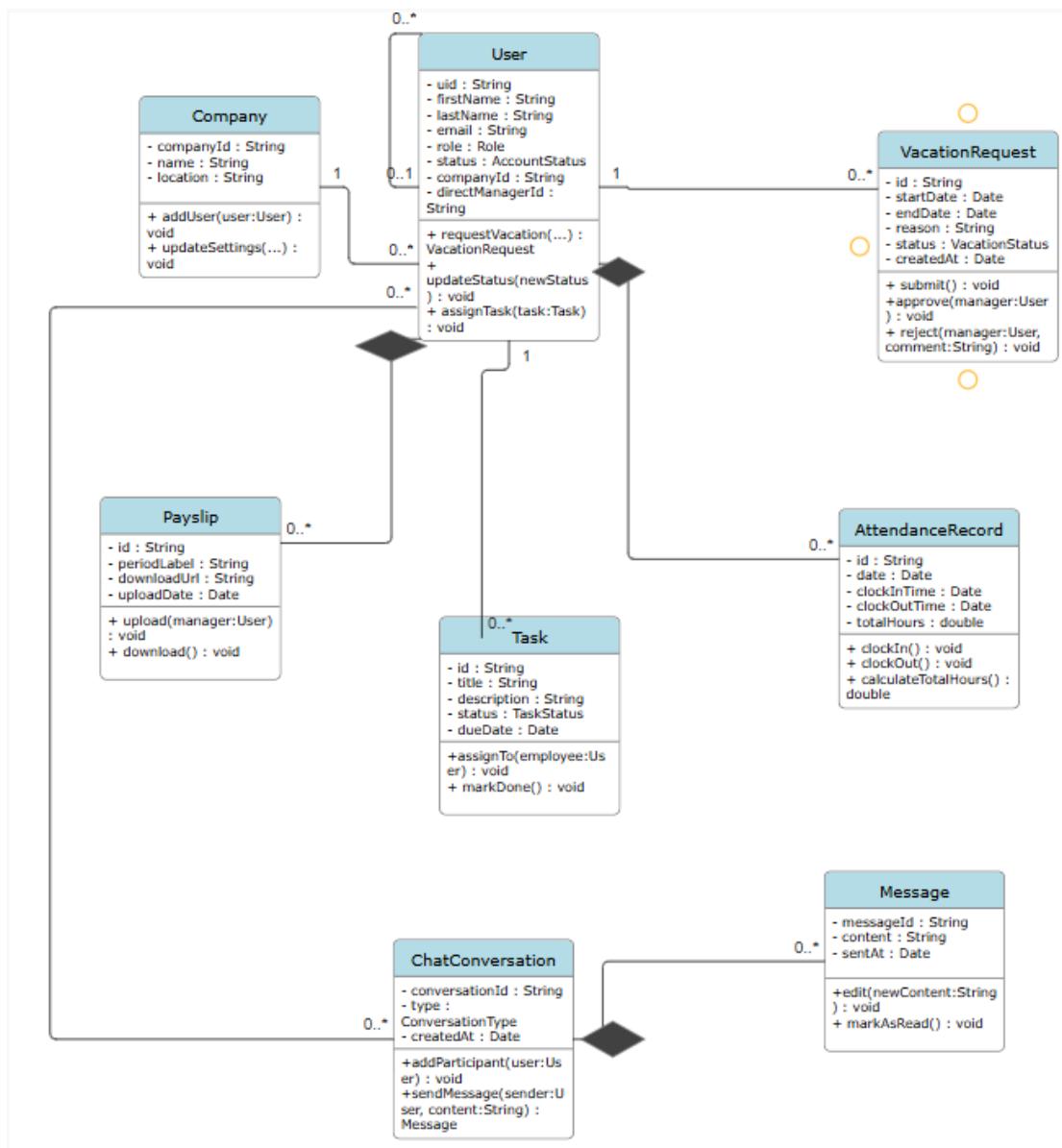
מטלה 3- מסמך ניתוח

מגישים:

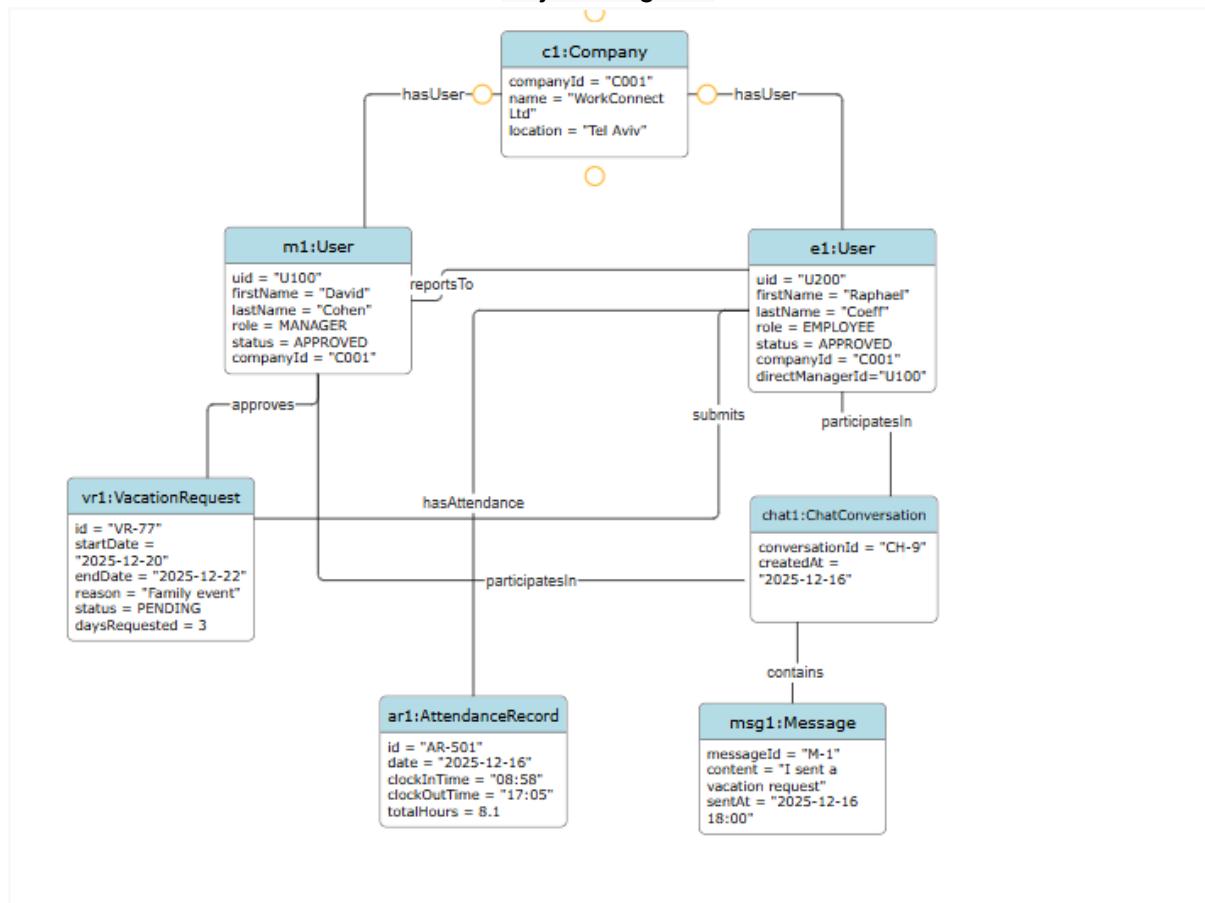
רפאל קואפיק 337614747
שי קרונפלד 322234782
משה אסקרוב 314085986

חלק א'- דיאגרמות

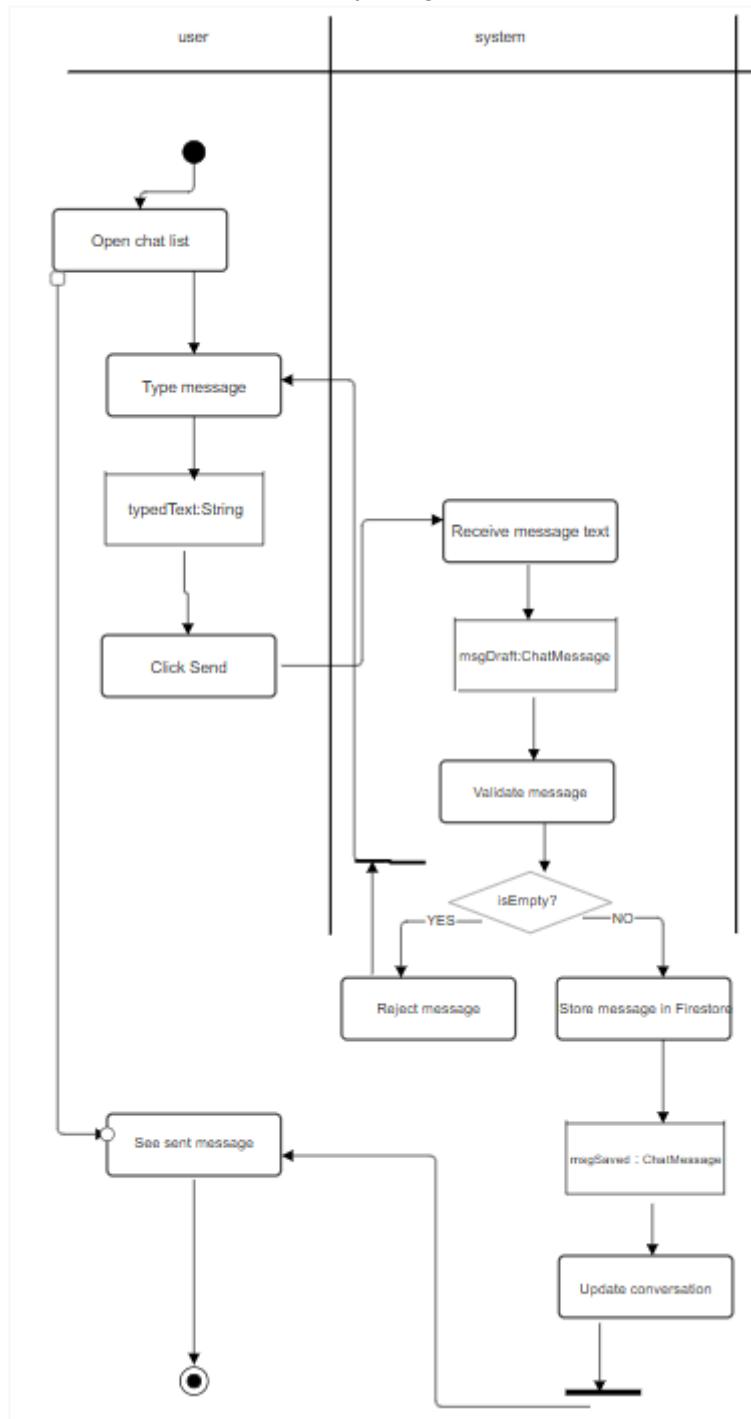
Class Diagram:



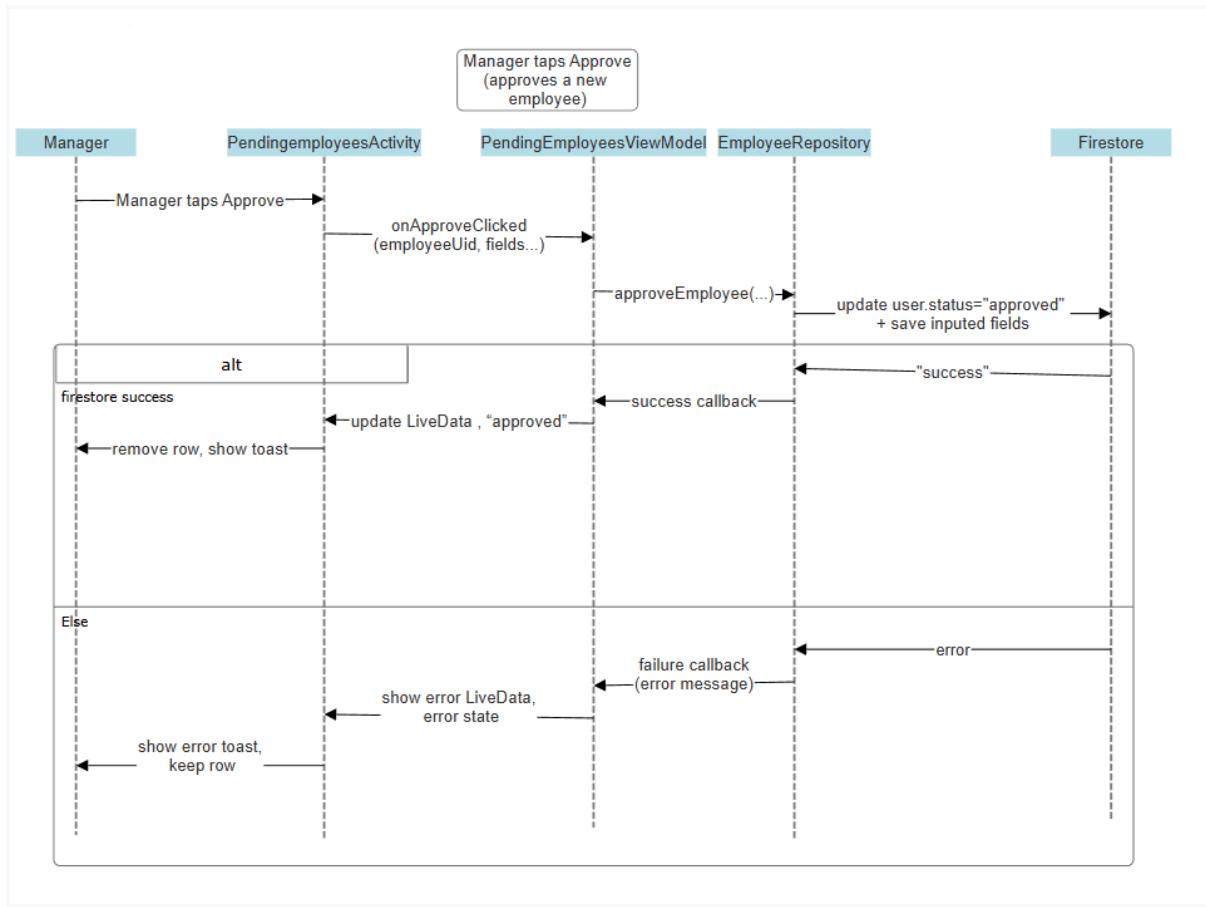
Object Diagram:



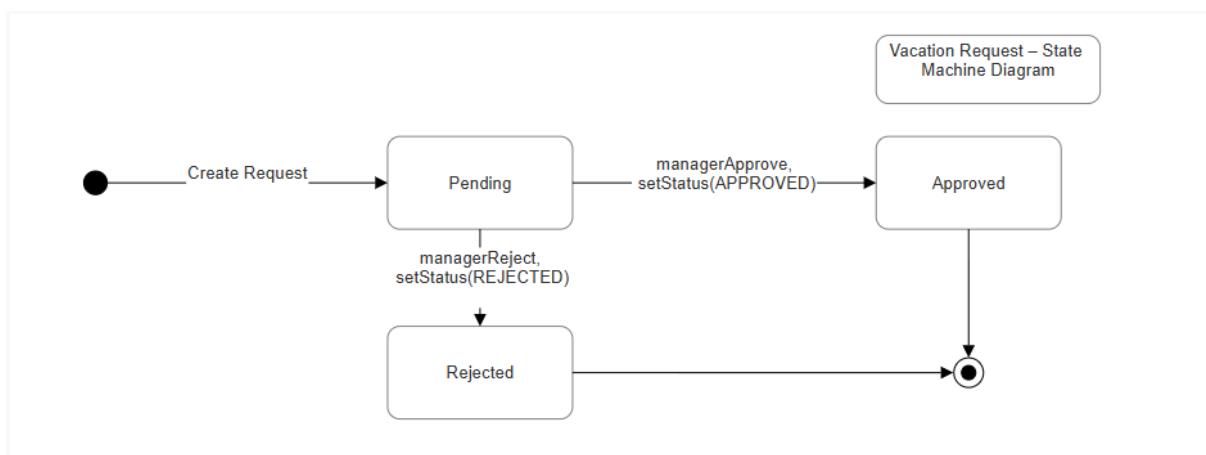
Activity Diagram:



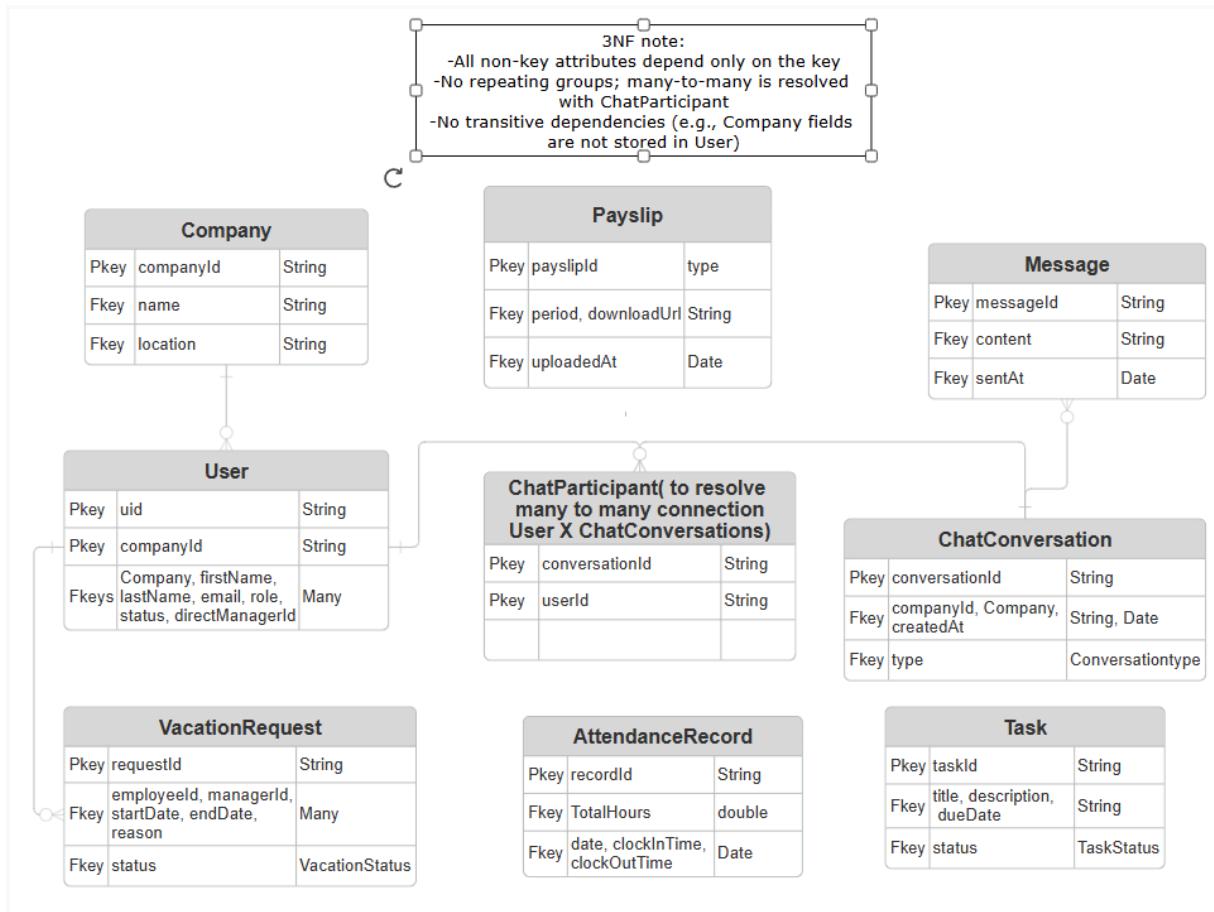
Sequence Diagram:



State Machine Diagram:



ERD (Entity-Relationship Diagram):



חלק ב' - מבנה מסכי המערכת

רצינו להבהיר לפני פירוט חלק זה מספר נקודות:

- יש במערכת מסכים שימושיים גם לעובדים וגם מנהליים. הסיבה לכך היא שמנהל הוא גם עובד, ולכן אופן השימוש והזרימה במסכים האלה זהים. כדי להימנע מchezנות, תיארנו את המסכים והזרימה המשותפים רק בחלק של העובדים. בחלק של המנהליים פירטנו רק את המסכים והפעולות שייחודיים למנהל.

- בפרק הזרימה תיארנו בצורה מפורטת תהליכיים שכבר התחלנו למשב בפועל באפליקציה. לעומת זאת, תהליכיים שעדיין לא מומשו תוארו באופן כללי יותר, כדי להסביר את הרעיון והכיוון שאליו אנחנו מתכוונים להגעה בהמשך הפיתוח.

1. רשימת מסכים מרכזיים:

• מסכים להרשמה (גם לעובדים וגם למנהל)

- Login (כניסה)
- Register Type (בחירה סוג הרשמה)
- Register Company (רישום חברת חדשה – מנהל מציף חברת למערכת)
- Register Employee (רישום עובד חדש)

• מסכים עבור כלל העובדים-

- מסך בית לעובד Employee Home Activity
- מסך my Vacations Activity (רישימת החופשות של העובד)
- מסך New Vacation Request Activity (בקשה חופשה חדשה)
- מסך my Shifts Activity (רישימת המשמרות של העובד)
- מסך Add Availability To Shifts Activity (הגשת זמינות משמרות)
- מסך My Salary Slips Activity (תלאושי שכר שלי)
- מסך chat (צ'אט)
- מסך video call (שיחות וידאו)

• מסכים נוספים עבור המנהלים-

- מסך בית למנהל Manager Home Activity
- מסך Pending Employees Activity (עובדים שמתינו 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 תלאושי שכר)

2. קשרים וזרימת נתונים בין מסכינים:

• זרימה ראשית - התחברות והרשמה:

- האפליקציה נפתחת - LoginActivity
- אם המשתמש כבר מחובר, יכנס מייל וסיסמה וילחץ על Login
- בבודק את ה- role מה- Firestore
 - LoginActivity → ManagerHomeActivity אם manager
 - LoginActivity → EmployeeHomeActivity אם employee
 - אם ילחץ על כפתור Register At RegisterTypeActivity

- RegisterTypeActivity
- אם ילחץ על Create new company At Create new company
- RegisterTypeActivity → RegisterCompanyActivity
 - אם ילחץ על Join existing company At Join existing company
 - RegisterTypeActivity → RegisterEmployeeActivity
 - אם ילחץ על Back יחזיר חזרה למסך הקודם. כלומר:
 - RegisterTypeActivity → LoginActivity

- RegisterCompanyActivity
- מילוי פרטי חברה + מנהל.
- אם ילחץ על Create company
 - יצירת משתמש מנהל + מסמך חברה וגם ManagerHomeActivity
 - אם ילחץ על Back יחזיר חזרה למסך הקודם. כלומר:
 - RegisterCompanyActivity → RegisterTypeActivity

- RegisterEmployeeActivity
- מילוי שם מלא, אימייל, סיסמה, קוד חברה.
- אם ילחץ על Create employee account
 - יצירת משתמש עובד עם status = pending
 - Toast: הרשמה ממتنינה לאישור מנהל.
 - וכן, RegisterEmployeeActivity → LoginActivity כי התהליך הסתיים כרגע.
- אם ילחץ על Back יחזיר חזרה למסך הקודם. כלומר:
 - RegisterEmployeeActivity → RegisterTypeActivity

• זרימה באזן העובד

- אם הגיע ל- EmployeeHomeActivity אחרי LoginActivity

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

.Logout → ('LoginActivity

- אם הגיע ל- myAttendanceActivity
 - העובד רצה לסמן שהגיע לעבודה/ יצא ממנה או שרצה לראות את היסטוריית הנווכחות שלו.
 - עם פתיחת המסך מוצג לוח שנה בפורמט חודשי כאשר היום הנוכחי יהיה בולט ויזגן נתוני כניסה יציאה שלו מאותם יום.
 - אם ילחץ על entry / exit תדוח כניסה/ יציאה בהתאם ותציג לו התראה על השעה שנרשמה.
 - כפותור כניסה או יציאה יקבע בהתאם לחיצת הקודמת שנלכדה בנווכחות.
 - כפותור Back יחזיר חזרה למסך הקודם. ככלומר:
myAttendanceActivity → EmployeeHomeActivity

- אם הגיע ל- MyShiftsActivity
 - העובד רצה לראות את המשמרות שלו.
 - עם פתיחת המסך מוצג לוח שנה חודשי, כאשר היום הנוכחי מודגש באופן ברור.
 - כבירית מחדל יציג המשמרות אליון העובד משובץ ביום הנוכחי.
 - אם ילחץ על יום אחר בלוח השנה, יציגו לעובד המשמרות אליון הוא משובץ בתאריך שנבחר.
 - אם ילחץ על month set availability for this month אז העובד רצה להזין זמינות חדשה וילק: MyShiftsActivity → AddAvailabilityToShiftsActivity
 - כפותור Back יחזיר חזרה למסך הקודם. ככלומר:
MyShiftsActivity → EmployeeHomeActivity

- אם הגיע ל- AddAvailabilityActivity
 - העובד רצה 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 העדפה למשרת (/ NOT CAN / PREFER).(CAN'T).
 - בחולניות יש כפותור cancel שאם המשתמש ילחץ עליו החולנית תיסגר ללא שמירה.
 - אם ילחץ על DONE השינויים ישמרו.

- אם הגיע ל- MyVacationsActivity
 - יציג רשימת בקשות החופשה של המשתמש.
 - אם ילחץ על כפותור New Vacation, העובד רצה לפתח בקשה לחופשה חדשה וילק: MyVacationsActivity → NewVacationRequestActivity
 - כפותור Back יחזיר חזרה למסך הקודם. ככלומר:
MyVacationsActivity → EmployeeHomeActivity

- אם הגיע ל- NewVacationRequestActivity
 - העובד יבחר תאריכים לחופשה ויכתוב סיבת.
 - כפותור Send request
 - יוצר בקשה חדשה (VacationRequest) עם סטטוס pending.
 - Toast הצלחה.
 - חזרה ל- ()finish → ()EmployeeHomeActivity

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

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

- אם הגיעו ל-ChatsActivity
 - העובד רוצה לניהל שיחות עם משתמשים אחרים בחברה.
 - עם פתיחת המסך תוצג רשימת ציאים קיימים של העובד (לפי שיחות אחרונות).
 - אם העובד ילחץ על שיחה מסוימת, תיפתח תצוגת השיחה ובها יוצג ההודעות הקודומות.
 - העובד יוכל להקליד הודעה וללחוץ Send, ההודעה תישלח ותופיע מיד בשיחה.
 - אם ילחץ על יצירת שיחה חדשה (New Chat), יוכל לבחור משתמש מהרשימה ולהתחליל שיחה חדשה.
- כפטור Back יחזיר חזרה למסך הקודם. כלומר:
. ChatsActivity → EmployeeHomeActivity

- אם הגיעו 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.
- כפטור Back יחזיר חזרה למסך הקודם. כלומר:
. VideoCallActivity → EmployeeHomeActivity

- אם הגיעו ל-MyProfileActivity
 - העובד רוצה לצפות בפרטים האישיים שלו.
 - עם פתיחת המסך יוצג פרטי העובד כמו: שם מלא, אימייל, תפקיד, שם החברה וכו' וכן יוצג איזור "יעדי לתלושי השכר".
 - בחלק של תלושי השכר תוצג רשימה לפי חודשים.
 - אם העובד ילחץ על תלוש שכר מסויים תיפתח אופציה של הציג או הורדה של התלוש.
- כפטור Back יחזיר חזרה למסך הקודם. כלומר:
. MyProfileActivity → EmployeeHomeActivity

• **זרימה בצד המנהל**

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

■ **כפתורים בניווט:**

● **אישיות:**

- . 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 ○

● **ניהול:**

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

• **חזרה למסך ההתחברות (LoginActivity) →**

○ אם הגיע ל- PendingEmployeesActivity

■ **מציג RecyclerView של עובדים חדשים שסמכותינו לאישור:**

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

■ **אם הגיע ל- dialogApproveEmployee**

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

■ **אם הגיע ל- PendingVacationRequestsActivity**

- מוצג RecyclerView של רשימת בקשות לחופשות של העובדים שסמכותינו לאישור.
- מתוך רשימת הבקשות יכול להזich על:
- כפתור Reject / Approve לכל בקשה.
- אם ילחץ על כפתור Back יחזיר חזרה למסך הקודם. כלומר:
- . PendingVacationRequestsActivity → ManagerHomeActivity

■ **אם הגיע ל- ManageAttendanceActivity**

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

- המנהל יוכל לבחור עובד מהרשימה, ולאחר הבחירה יוצגו נתוני העובד הנוכחות של העובד הנבחר בפורמט של רשימה לפי תאריכים עם סעיפים נוכחות.
- המנהל יוכל לבחור יום מסוים מתוך רשימת הימים הנוכחיות של העובד כדי לצפות בפרטים מלאים של אותו יום.
- המנהל יוכל לעדכן נתונים כניסה ויציאה של עובדים וכמם לאשר דיווח 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.
- בעט פתיחת המסך יוצגו למנהל הקבוצות אותן הוא צריך לשבץ. בעט לחיצה על קבוצה מסוימת יפתח לוח שנה בתצוגה חדשית.
- בעט לחיצה על תאריך מסוים המשתמש ילחץ על כפתור ASSIGN וויאציג לו המשמרות של אותו יום.
- אם ילחץ על משמרת מסוימת, תוצג לו רשימה שכותבים 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 יחזיר חזרה למסך הקודם. כלומר:
 - . ManageShiftsActivity → ManagerHomeActivity

■ אם הגיע ל- UploadSalarySlipsActivity ■

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

■ אם הגיע ל- CompanySettingsActivity ■

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

3. ייצוג חסויות:

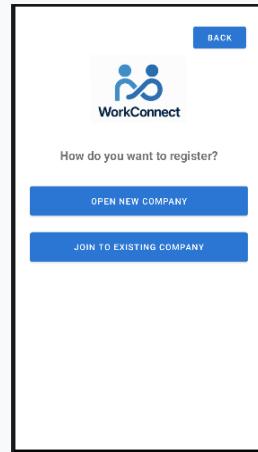
מוכי התקשורת והרשמה:

:Register Company מוק



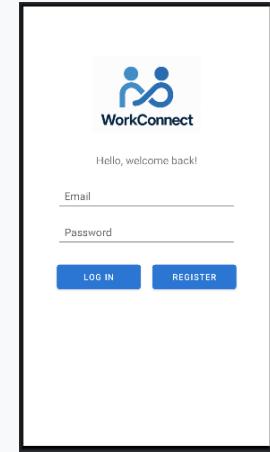
The registration screen for a new company. It features the WorkConnect logo at the top left. A blue 'BACK' button is at the top right. Below the logo is a button labeled 'Open new company'. The form consists of four input fields: 'Company name', 'Your full name', 'Email', and 'Password (min 6 chars)'. At the bottom is a blue 'CREATE COMPANY AND ACCOUNT' button.

: Register Type מוק



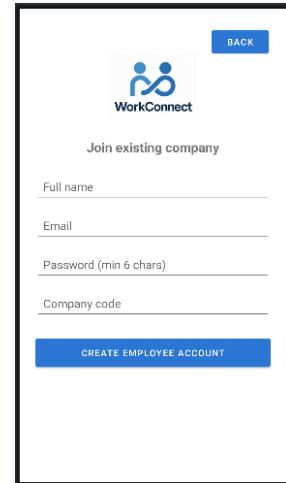
The registration screen for selecting a company type. It features the WorkConnect logo at the top left. A blue 'BACK' button is at the top right. Below the logo is the text 'How do you want to register?'. There are two blue buttons: 'OPEN NEW COMPANY' and 'JOIN TO EXISTING COMPANY'.

: Login מוק



The login screen. It features the WorkConnect logo at the top left. The text 'Hello, welcome back!' is at the top right. Below it are two input fields: 'Email' and 'Password'. At the bottom are two blue buttons: 'LOG IN' and 'REGISTER'.

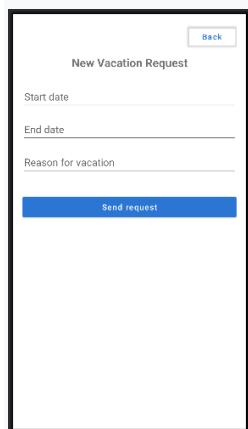
:Register Employee מוק



The employee registration screen. It features the WorkConnect logo at the top left. A blue 'BACK' button is at the top right. Below the logo is the text 'Join existing company'. The form includes four input fields: 'Full name', 'Email', 'Password (min 6 chars)', and 'Company code'. At the bottom is a blue 'CREATE EMPLOYEE ACCOUNT' button.

מוכן לעבוד -

:New Vacation Request מוכן לעבוד -



New Vacation Request

Start date _____

End date _____

Reason for vacation _____

Send request

:my Vacations מוכן לעבוד -

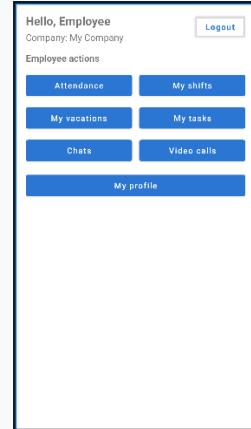


Vacation Date Status

Item 0
Item 1
Item 2
Item 3
Item 4
Item 5
Item 6
Item 7
Item 8
Item 9

New Request

:Employee Home מוכן לעבוד -



Hello, Employee
Company: My Company
Logout

Employee actions

Attendance My shifts

My vacations My tasks

Chats Video calls

My profile

:all chats מוכן לעבוד -



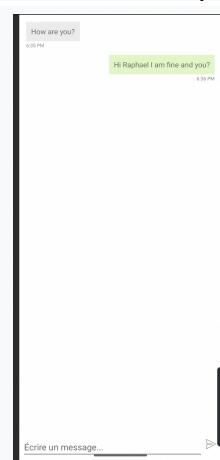
Search employee...

Raphael Coeffic
How are you?
Dec 16, 2025 5:03:08 PM

chat 0

Dec 16, 2025 2:07:49 PM

:chat מוכן לעבוד -

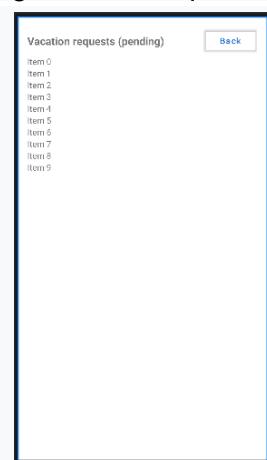


How are you?
6:01 PM

Hi Raphael I am fine and your?
8:11 PM

Ecrire un message... ➤

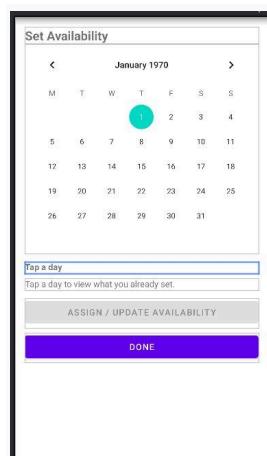
:Pending vacation request מוכן לעבוד -



Vacation requests (pending)

Item 0
Item 1
Item 2
Item 3
Item 4
Item 5
Item 6
Item 7
Item 8
Item 9

:add availability מוכן לעבוד -



Set Availability

January 1970

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

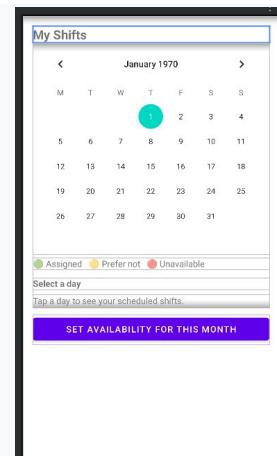
Tap a day

Top a day to view what you already set.

ASSIGN / UPDATE AVAILABILITY

DONE

:my shifts מוכן לעבוד -



My Shifts

January 1970

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Assigned Prefer not Unavailable

Select a day

Tap a day to see your scheduled shifts.

SET AVAILABILITY FOR THIS MONTH

:item pending employee מוקן :Pending Employees מוקן Manager Home מוקן

:item pending vacation request מוקן :dialog approve employee מוקן

:manage shifts מוקן

חלק ג' - מבנה הקוד ותבניות עיצוב עבור Android Studio

1. תיקיות הפרויקט שלנו הן:

• ן:

- בתיקייה זו נמצאים Activities/Fragments של המ██ים.
- תפקיד הקבצים בתיקייה זו: להציג מידע, לקבל קלט מהמשתמש, לנוט בין מסכים, ולהאזין ל- LiveData מה- ViewModel.
- הקבצים בתיקייה זו נמצאים בתוך יש 3 תת-תיקיות בשביל סדר:
 - תיקיית auth : בה יש הקבצים שאחראים 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: לשמר מצב מסך, לבצע לוגיקה שקשורה למ██, ולדבר עם ה- Repository.
- הקבצים בתיקייה זו נמצאים בתוך יש 3 תת-תיקיות בשביל סדר:
 - תיקיית auth : בה יש הקבצים שאחראים על רישום והתחברות.
 - תיקיית manager : בה יש את הקבצים שקשורים למנהל.
 - תיקיית employee : בה יש את הקבצים שקשורים לעובד.

:repository •

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

:models •

- בתיקייה זו יש אובייקטים שימושיים שמייצגים ישותה המערכת: User, Employee, Manager, Company, VacationRequest.

:adapters •

- מתאימים של RecyclerView: RecyclerAdapter (למשל: VacationRequestsAdapter, PendingEmployeesAdapter).
- תפקיד הקבצים בתיקייה זו: לחבר בין רשימת נתונים לבין תצוגת שורות.

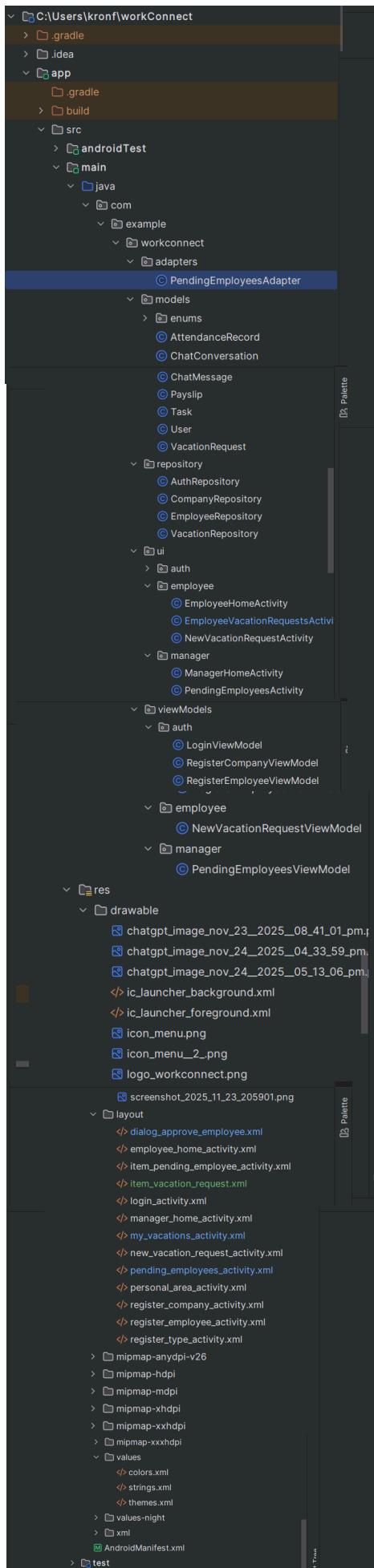
בחרנו דוקא במבנה זהה כי הוא מתאים הכי טוב לדרישות של הפרויקט שלנו ולסוג הבעיות שהוא פותר. המבנה נתן סדר ברור ומונע "בלאגן" בקוד שימושי פים עוד מ██ים ופיצ'רים.

• הפרדת אחירות ברורה:

- ונו- אחראי רק על תצוגה, ניוט וקלט מהמשתמש.
- viewModel- מנהל את מצב המסך ואת הלוגיקה שקשורה למ██ (למשל: טעינה, הודיעות שגיאיה, מתי להציג רשימה).
- models- מגדיר את מבני הנתונים (Company, User, VacationRequest וכו').
- זה חשוב כי ככה לא "מערבעים" קוד תצוגה עם קוד נתונים, והיכול נהיה קרייאן וקל לתחזוקה.

- התאמת טبيعית ל- MVVM (האררכיטקטורה שבחרנו):
 - MVVM עובד היטב כי טוב כשיש שכבות ברורות: Repository → Firebase
 - המבנה הזה ממש "מכירח" לעובוד נכון MVVM ומונע מצב שבו Activity מבצע קריאות שירות ל- Firestore (שזה מקור לבאגים וקיים מסורבל).
- יכולת גדרילה של הפרויקט בלי לשבור את הקיטי:
 - בפרויקט שלנו אנחנו צפויים להוסיף עוד מסכים ותהליכיים (לדוגמא: הגדרות חברה, מדיניות פרטיות, דוחות הנהלה, חוקים כמו מינוס ימי חופשה). וככה כל פיצ'ר חדש נכנס למקום קבוע:
 - UI למסכים
 - ViewModel ללוגיקה
 - Repository לננתונים
 - אפשר להוסיף בלי "למרוח" קוד בכל הפרויקט.
- עבודה נכונה עם Firebase ושינויים עתידיים:
 - Firebase הוא מקור נתונים חיצוני, יוכל לקרטות שימוש אחד נרצה להחליף חלק מהקריאות או לשנות מבנה מסמכים ב- Firestore. אם כל הגישה לננתונים מרכזת ב- Repository, אנחנו נשנה במקום אחד במקום אחר על כל המסכים.
- קל יותר לבדוק ולתקן תקלות:
 - למשל כשיש בעיה בבקשת חופשה:
 - אם זה בתצוגה אז מחפשים ב- ui.
 - אם זה בננתונים או בשאליתה אז מחפשים ב- repository.
 - אם זה במצב מסך (loading/error) אז מחפשים ב- viewmodel וזה קצר משמעותית זמן דיבוג.
- עקביות בין מודולים שונים:
 - המערכת שלנו מחולקת לפקדים. לכן יהיה לנו חשוב שגם ה- UI למשל יהיה מסודר לפי תחומים. זה עוזר להבין מהר איפה כל דבר נמצא ומונע כפליות.

צירוף צילום מסך של מבנה התיקיות בקוד:



2. בחרנו בתבנית עיצוב MVVM נפרט את יתרונות של בחירה דואק בתבנית זו:

- **תחזוקתיות:**
 - שינוי בולוגיקה של תהילך לא דורש שינוי במסכים עצמם.
 - הקוד מחולק לשכבות: `ViewModel` ↔ `Repository` ↔ `Firebase` ↔ `UI` ולכן קל יותר למצוא קבצים ולהבין "אייפה כל דבר נמצא".
- **סקליבליות:**
 - קל להוסיף פיצרים חדשים (למשל: סוג חופשות, דוחות למנהל, התראות) בלי "לשבור" קוד קיימ.
 - אפשר להוסיף מקור נתונים נוסף בעתיד בלי לשנות את המרכיבים - רק לעדכן את ה- `Repository`.
- **ביצועים וחווית משתמש:**
 - ViewModel שומר מצב ולא "מאבד מידע" בסיבובי מסך/שחזור Activity, וכך פחות קיראות חוזרת ל- `Firebase`.
 - אפשר להוסיף אופטימיזציה בשכבה ה- `Repository` בלי לגעת ב- `UI`.
 - אפשר Recycler View + Adapter מאפשרים הצגה יעילה של רשימות (כמו בקשנות חופשה/עובדים בהמתנה).
- **בדיקות ואיכות קוד:**
 - קל יותר לבדוק לוגיקה ב- `ViewModel` וב- `Repository` (יחידות בדיקה), כי היא לא תלואה ישירות במסך או ב- `UI` Android.
 - הפרדת אחריות מorigina באגים ומקרה על דיבוג.