**דו"ח גרסה איטרציה 2**

**קבוצה 2**

מגישים:

אלון גולומבק 302361647

ינון בן-זכרי 204439806

עידו עפרוני 204677777

איתי כץ 312178312

תומר זיינפלד 312543929

**A football player on a field

Description automatically generated**

תוכן עניינים

[תיעוד ניהול גרסה 3](#_Toc38128619)

[בעלי תפקידים 3](#_Toc38128620)

[הספקים 3](#_Toc38128621)

[משימות מהאיטרציה הקודמת 3](#_Toc38128622)

[מימוש שכבת השירות 3](#_Toc38128623)

[מימוש שכבת התחום 3](#_Toc38128624)

[מימוש של שכבת המידע 4](#_Toc38128625)

[מימוש ה-UC 4](#_Toc38128626)

[בדיקות 6](#_Toc38128627)

[ניהול משימות, תיעדוף ושיוך משימות 7](#_Toc38128628)

[תהליך הקדם 7](#_Toc38128629)

[ממשק הניהול 8](#_Toc38128630)

[תקשורת בין חברי הקבוצה, תיעוד קבצים, קישורים, סיכומי פגישה ועדכונים 10](#_Toc38128631)

[ניהול גרסאות 12](#_Toc38128632)

[ניהול בדיקות 12](#_Toc38128633)

# תיעוד ניהול גרסה

## בעלי תפקידים

* מנהל גרסה: ינון
* אחראי בדיקות: תומר
* מתכנתים: איתי, עידו, אלון, ינון ותומר.

## הספקים

### משימות מהאיטרציה הקודמת:

* תיקונים בתרשים המחלקות
* תיקונים בתרשים ארכיטקטורה
* תיקונים במסמך הדרישות, לדוגמה: מספור, תוכן עניינים.
* עדכון מילון המונחים.

### מימוש שכבת השירות

הגורם המקשר בין שכבת המשתמש לבין ה-Business Layer. מהווה מעין חוצץ בין ה-UI לבין שכבת התחום. שכבה זו מחולקת למחלקות שירות, כל מחלקה כזאת תפקידה לקבל קלט מהמשתמש בצורה של מחרוזות, ולהעבירם אל שכבת התחום. שכבת התחום מבצעת את העיבוד על הקלט, ואל שכבת השירות חוזרת תשובה האם הפעולה בוצעה בהצלחה או לא. רק לשכבת השירות יש את היכולת לדבר עם ה-Handlers בשכבת ה-Business והיא איננה מודעת לתהליך שנעשה על הקלט שהמשתמש הכניס. מכאן שלכל Handler יש מחלקת שירות משלו: System Service, Team Service, League Service ו-Match Service.

### מימוש שכבת התחום

מהווה את לב המערכת - Business Layer. מכיל את האובייקטים הקונקרטיים ואת הפונקציונליות שלהם. הפונקציות של האובייקטים נקראות מתוך Handlers, כל Handler אחראי על תחום אחר כדי למנוע עומס של אחריות, ובאחריותו לדבר עם שכבת השירות (שעליה נפרט בהמשך). ה-Handlers שמימשנו הם:

* System Controller – אחראי על התפקוד השוטף של המערכת. תפקידו לדבר עם האובייקטים שקשורים לאישורים, משתמשי מערכת, החלפת סיסמאות, אתחולים ופניות ל-Data Layer.
* Team Controller – אחראי על ניהול המידע הקשור לקבוצות, כגון שינוי סטטוס קבוצה, הוספת/מחיקת מנהלי קבוצה ובעלי קבוצה, דיווח על הכנסות והוצאות הקבוצה ועוד.
* League Controller – אחראי על ניהול המידע הקשור לליגות, כגון השופטים הנוכחיים בליגה, מינויים של חברי ההתאגדות, ניהול אצטדיונים ועוד.
* Match Controller – אחראי על ניהול המידע של המשחקים השונים. כל משחק מכיל את המידע על עצמו וה-Controller מנהל את המידע הזה, למשל מינוי שופט למשחק.

### מימוש של שכבת המידע

שכבה אשר באחריותה לשמש כבסיס נתונים. בגרסה זו נוצרה מחלקה בשם DemoDB שתוחלף בעתיד בבסיס נתונים חיצוני שאינו יהיה חלק מזיכרון המערכת. שכבה זו היוותה מאגר נתונים שבו נעזרנו כדי לממש את הבדיקות של המערכת. מטרת המחלקה הייתה לוודא כי שום אובייקט אינו מסתמך על יכולת לשמור נתונים בזיכרון של התוכנית בזמן ריצה ולדמות את התקשורת עם בסיס הנתונים כדי למצוא ולעדכן מידע.

## מימוש ה-UC

פירוט נרחב יותר לכל מימוש נמצא בממשק ניהול המשימות של הצוות שיפורט בהמשך.

#### מימוש באיטרציה הנוכחית

להלן רשימה של ה-UC שמומשו באיטרציה זאת:

|  |  |  |
| --- | --- | --- |
| אובייקט | מספר | פירוט קצר |
| מערכת | 1.1 | אתחול מערכת |
| אוהד | 3.4 | הגשת תלונה |
| בעל קבוצה | 6.1 | ניהול נכסים |
| 6.2 | מינוי בעל קבוצה |
| 6.3 | הסרת מינוי של בעל קבוצה |
| 6.4 | מינוי מנהל קבוצה |
| 6.5 | הסרת מינוי של מנהל קבוצה |
| 6.6 | סגירת קבוצה |
| 6.7 | דיווח הכנסות והוצאות של קבוצה |
| מנהל קבוצה | 7.1 | ביצוע פעולות ניהול בהתאם להרשאות |
| מנהל מערכת | 8.1 | סגירת קבוצה לצמיתות |
| 8.2 | הסרת מנויים בהתאם לאילוצי המערכת |
| 8.3 | צפיה ותגובה לתלונות במערכת |
| נציג ההתאחדות | 9.1 | הגדרת ליגה |
| 9.2 | הגדרת עונה עפ"י שנה |
| 9.3 | מינוי והסרת שופטים |
| 9.3.1 | הזמנה עבור השופט להשתתף בליגה |
| 9.4 | שיבוץ שופטים בעונה |
| 9.5 | הגדרת מדיניות חישוב ניקוד ומיקום בליגה |
| 9.6 | הגדרת מדיניות לשיבוץ משחקים |
| שופט | 10.2 | צפיה במשחקים אליהם הוא משובץ |
| 10.3 | עדכון אירועים במהלך משחק |
| 10.4 | עדכון אירועים לאחר שמשחק הסתיים ע"י שופט ראשי |

לסיכום, המימושים אשר באו לידי ביטוי בגרסה זו (כולל מבחני קבלה):

* אתחול המערכת.
* ניהול קבוצות.
* אתחול עונה.

#### מימוש באיטרציות הבאות

דרישות שימומשו באיטרציות הבאות:

|  |  |  |
| --- | --- | --- |
| אובייקט | מספר | פירוט קצר |
| אורח | 2.1 | הגדרת משתמש שנכנס למערכת כאורח |
| 2.2 | רישום למערכת |
| 2.3 | כניסה למערכת באמצעות פרטים מזהים |
| 2.4 | צפיה במידע על פריטים במערכת |
| 2.5 | חיפוש |
| אוהד | 3.1 | יציאה מהמערכת |
| 3.2 | מעקב אחר דפים אישיים |
| 3.3 | רישום להתראות על משחקים |
| 3.5 | צפיה בהיסטוריית חיפושים |
| 3.6 | צפיה ועריכת פרטים מזהים אישיים |
| שחקן | 4.1 | עדכון פרטים אישיים |
| 4.2 | העלאת תוכן לדף אישי |
| שופט | 5.1 | עדכון פרטים אישיים |
| 5.2 | העלאת תוכן לדף אישי |
| מנהל מערכת | 8.4 | צפיה בלוגים |
| 8.5 | הפעלת בניית מודל ההמלצה |
| מערכת המלצה | 11 | מימוש מערכת המלצה |
| התראות | 12 | מימוש מערכת התראות |
| דרישות ברמת שירות – אבטחה, חווית משתמש | | |

### 

### בדיקות

במהלך כתיבת הקוד נכתבו בדיקות שכללו טסטים מקיפים על המימושים הקשורים לנושאים שפורטו בסעיף הקודם.

הטסטים שלנו הסתמכו על בסיס נתונים שבנינו מראש בתור מחלקה הנקראת DataBaseValues. תפקיד מחלקה זאת היא לדמות בסיס נתונים עבור הבדיקות. המחלקה משתמשת באובייקט מסוג DemoDB אשר משתמש בפונקציונליות המדמה בסיס נתונים. הטסטים בנויים משלושה סוגים:

|  |  |  |
| --- | --- | --- |
| סוג בדיקה | פירוט | סימון בטסטים |
| בדיקות יחידה | בדיקות אשר בודקות את הפונקציונליות של המחלקות הנמצאות בשכבת התחום בעזרת שימוש בפונקציות האטומיות שלהן. | UT\_TestName |
| בדיקות שילוב | בדיקות אשר משלבות כמה בדיקות יחידה לבדיקת עבודה משותפת של פונקציונאליות בין כמה מחלקות שונות. | IT\_TestName |
| בדיקות קבלה | בדיקות אשר מטרתן לבדוק את תפקוד המערכת מנקודת המבט של השכבת השירות. כל בדיקה כזו יוצרת אובייקט השייך ל-Service Layer ושולחת לפונקציות שלה קלטים בצורה אוטומטית ומדמה מצב בו משתמש מכניס קלט דרך ממשק המשתמש, שבגרסה זו עדיין אינו קיים ולכן יש לדמות אותו בעזרת הטסטים. מהלך העבודה עם בדיקות הקבלה יפורט בהמשך הדו"ח ויכיל הסברים על תהליך העקיבות. | UC\_TestName |

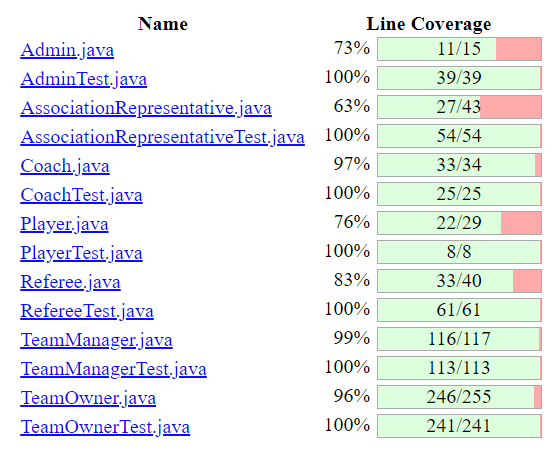
## PIT

הבדיקות נלוו בבדיקת מוטציות בעזרת המערכת PIT. המטרה בבדיקות אלה הן לבדוק מה מצב כיסוי השורות בקוד שלנו בהתאם לבדיקות, וכן בדיקה האם הקוד יודע להתמודד עם שינויים שהמערכת מבצעת בו.

העבודה עם התוכנה עזרה למקד את הטסטים בשורות קוד שלא כוסו וכן לטפל בתקלות שהתגלו באמצעות השינויים בקוד. הדו"ח מצורף כנספח למסמך זה.

מידע נוסף על ניהול הבדיקות מצוין בהמשך הדו"ח תחת הכותרת "ניהול בדיקות".

לאחר בניית הטסטים מריצים את בדיקות המוטיציה, וניתן לראות מה רמת הכיסוי של הטסטים:



כיסוי שורות

מספר תובנות שעלו שחלק מהשורות שלא כוסו נובע ממחלקות שטרם מומשו במלואן באיטרציה זאת.

## ניהול משימות, תיעדוף ושיוך משימות

### תהליך הקדם

הצוות התכנס בתחילת האיטרציה וקרא את הדרישות ביחד. לאחר מכן, התחלנו לחלק את האיטרציה למשימות על. בהמשך, חבר צוות האחראי על ניהול המשימות חילק כל משימת על לתתי משימות באמצעות ממשק שנפרט עליו בהמשך.

בתחילת האיטרציה, הצוות קיים ישיבה אחת לשלושה ימים כדי להתעדכן על ההתקדמות במשימות. כאשר הגענו לשלב כתיבת הקוד קיימנו ישיבה בסוף יום העבודה בה שילבנו גרסאות, עדכנו בהספקים וחילקנו משימות ליום הבא.

במהלך היום אם היו שינויים או עדכונים קיימנו ישיבה בהתאם לצורך או עדכנו את האחראי מהצוות על ניהול המשימות.

### ממשק הניהול

ניהול המשימות בצוות נעשה באמצעות הממשק בתוכנה Asana.

בממשק זה ניתן לקבוע לכל משימה עדיפות, חבר צוות אחראי, תוכן ואף להכניס תתי משימות קטנות יותר למשימת העל.

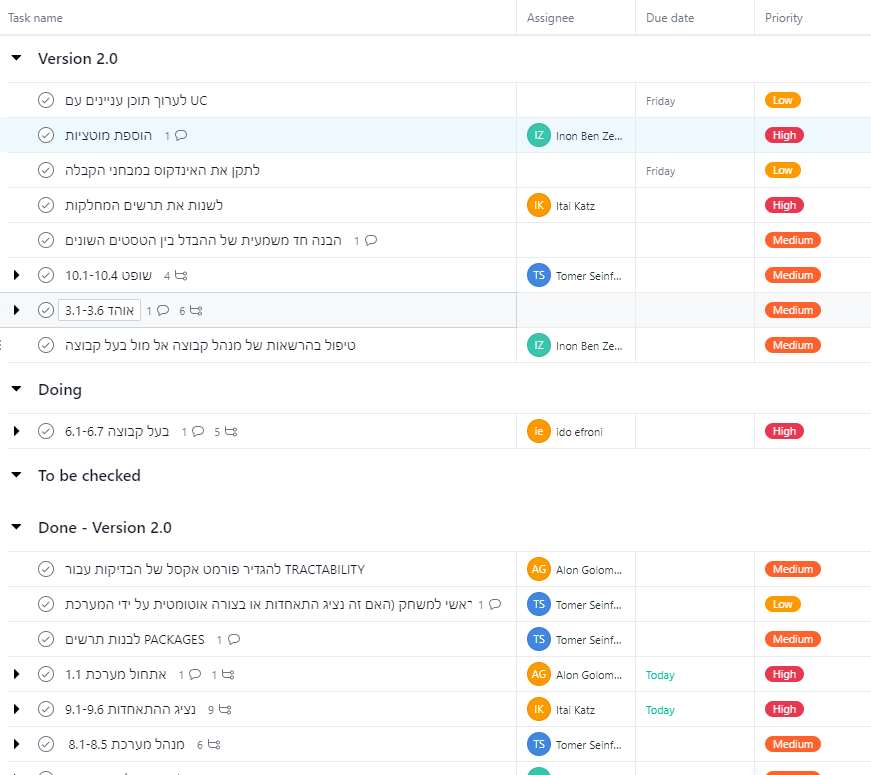
קישור: [https:/app.asana.com/0/1166532411538937/1166532411538937](https://app.asana.com/0/1166532411538937/1166532411538937)

ניתן לחלק את המשימות ב-Asana לפי קטגוריות. צורת העבודה שלנו מחולקת ל-4 שלבים:

1. שיוך לאיטרציה – מדובר במשימות שטרם בוצעו, אך שייכות לתחום האיטרציה.
2. Doing – משימות שעובדים עליהן.
3. To Be Checked – ישנן משימות הדורשות מעבר של חבר צוות נוסף, על כן לפני שהן יועברו לקטגוריה Done, הן יועברו לקטגוריה המוזכרת.
4. Done – משימות שבוצעו במסגרת האיטרציה שעליה אנו עובדים.

להלן תהליך הפיתוח לפי ציר זמנים:

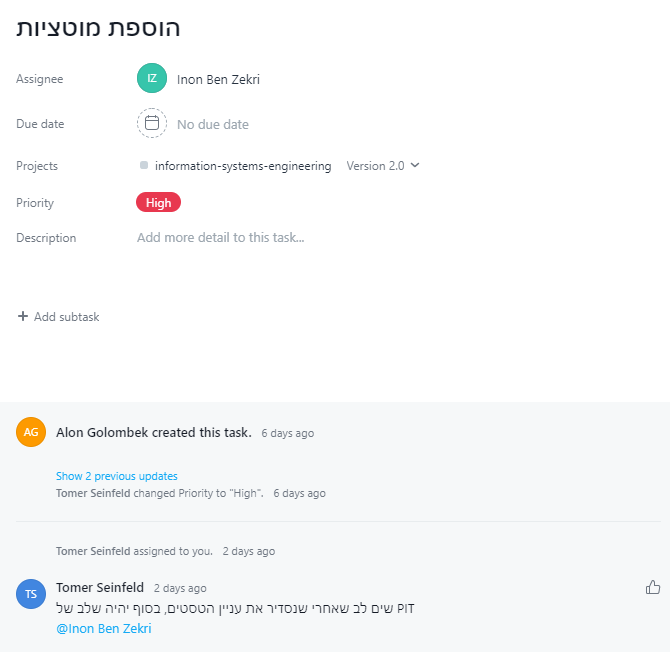
שיוך לאיטרציה 🡨 doing 🡨 to be checked 🡨 Done.



מסך ראשי ב-ASANA

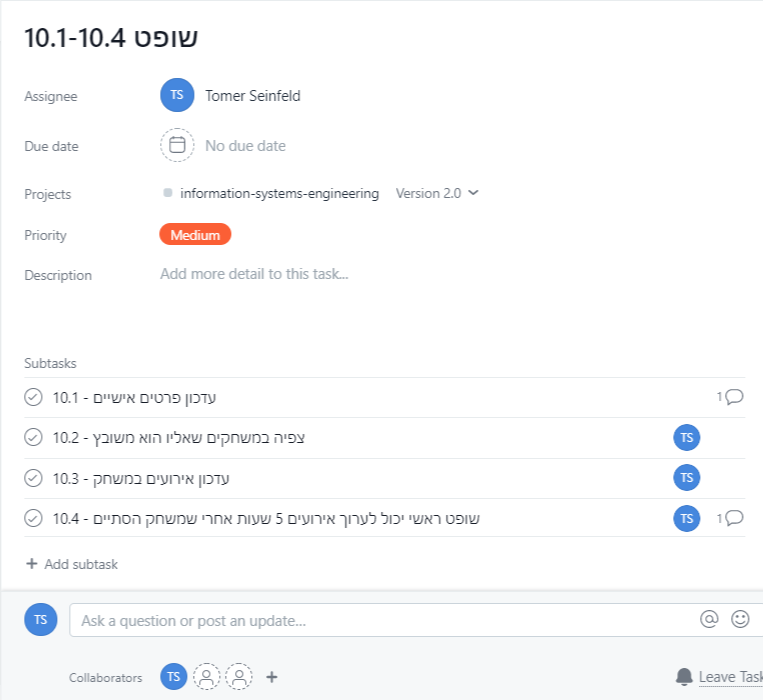
ב-Asana ניתן לשקף לשאר חברי הצוות את ההתקדמות במשימה:

בתוך כל דף משימה, אנו מגיבים, נותנים הערות/דגשים למבצעי המשימה. כמו כן, מבצעי המשימה יכולים לתקשר ביניהם על המשימה דרך דף המשימה או להגדיר תחומי אחריות ברורים יותר על המשימה בדף עצמו.



שיקוף חבר צוות בתהליך

כדי להתנהל בצורה הטקטית ביותר בחלוקת המשימות והדרישות בעבודה, אנו רוצים שהמשימות הניתנות יהיו כמה שיותר אטומיות. ב-Asana ישנה אפשרות לגזור תתי משימות, ולשבץ אותן לחברי הצוות.



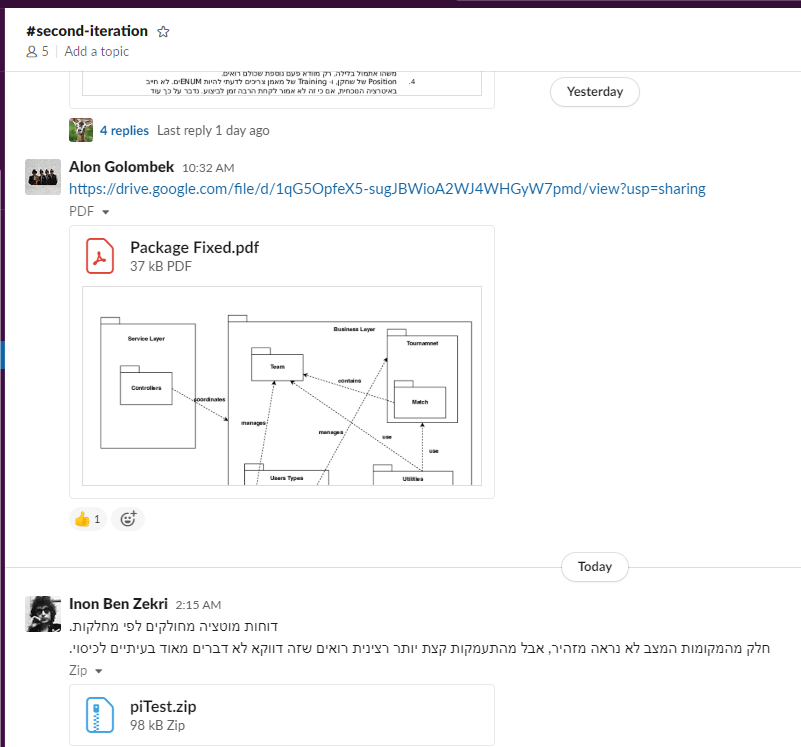
גזירת תתי ממשימות ממשימת על

## תקשורת בין חברי הקבוצה, תיעוד קבצים, קישורים, סיכומי פגישה ועדכונים

מלבד התקשורת השוטפת בין חברי הקבוצה ב-zoom ובעבודה המשותפת, אנו מתעדים את הנ"ל ב-slack על מנת לנהל תקשורת שקופה בין חברי הקבוצה, ועל מנת לתעד את חומרי העזר השונים אותם אנו צריכים עבור הפרויקט.

קישור: <http://team2informat-bjg8910.slack.com>

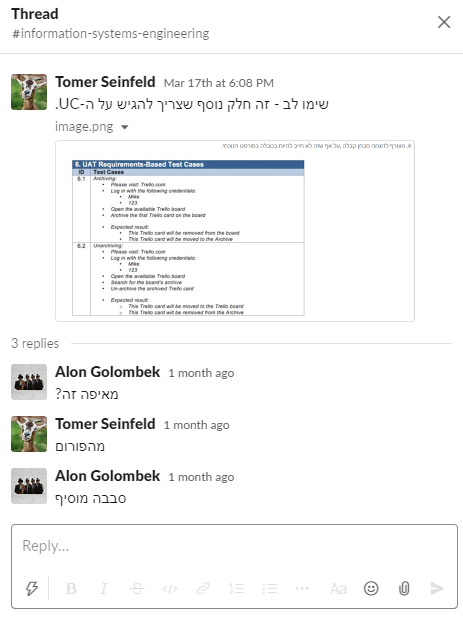
דף הפרויקט שלנו מחולק לפי ערוצים, כאשר כל ערוץ מייצג תחום עיסוק במסגרת הפרויקט. לדוגמה: ערוץ עבור כל איטרציה, "Traceability" , SSD וכן ערוץ כללי של הפרויקט. כל ערוץ משרת מטרה אחרת לתקשורת בין חברי הקבוצה.



ריכוז התוצרים בממשק

קבצים אשר דורשים גישה ועדכון עבור כל חברי הצוות נמצאים ב-GoogleDrive על מנת שכולם יוכלו לצפות ולערוך בהתאם לצורך את הקבצים באופן מקוון.

גם כאן ניתן להגיב על כל אחד מהפוסטים ולנהל דיון על הנושא שהועלה, כך שנוצר שיח מלא ושקוף בין חברי הקבוצה.



דיונים על תוצרים

## ניהול גרסאות

אנו משתמשים ב-github על מנת לתאם גרסאות קוד בפרויקט.

קישור: <https://github.com/yinonbz/FootballProject>

ה-repository של github מסונכרן עם ה-intellij של כולנו, וכל אחד עובד על ענף נפרד על מנת ליצור כמה שפחות קונפליקטים וכדי ליצור אי תלות בין הגרסאות השונות.

בכל ערב אנו עושים איחוד של הגרסאות, כדי שתהיה בקרה על התהליך ע"י כל חברי הצוות.

## ניהול בדיקות

בגרסה זו, בדיקות היו חלק בלתי נפרד ממימוש תשתית המערכת. כדי למנוע בעיות במימושים עתידיים אשר מתבססים על תשתיות אלו, ביצענו בדיקות רבות. כתיבת הבדיקות הותאמו לתרחישי הקבלה שנכתבו באיטרציה הראשונה.

כדי לשמור על אחידות בין חברי הקבוצה, בנינו פורמט אחיד שיעזור לנו לעקוב אחרי הבדיקות קבלה בצורת מטריצה.

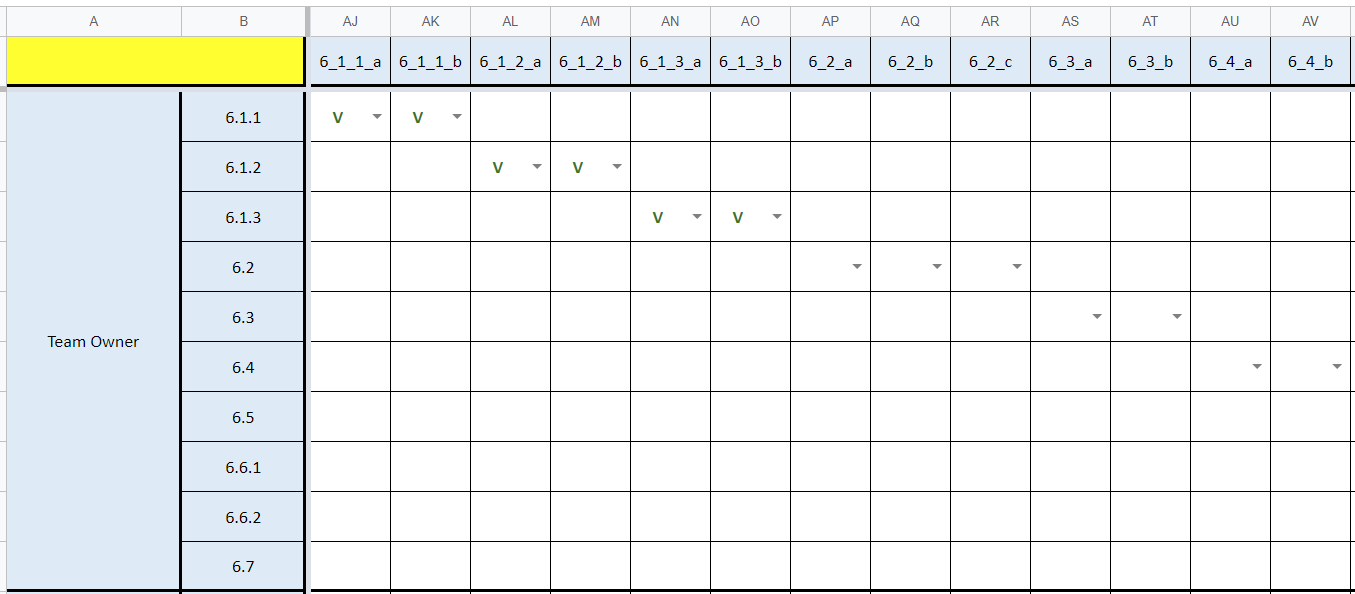
כל מבחן קבלה קיבל Test-ID חד ערכי, אשר הותאם גם למטריצה וגם לקוד, על מנת שנוכל לאתר ולעקוב אחר כל בדיקה ודרישה בצורה מהירה ומדויקת.

[קישור](https://drive.google.com/file/d/1LZdTlUtTWTnfKFKJCa-BKWANAtQmLnsh/view?usp=sharing) ל-Traceability Matrix שאיתו עבדנו.

כמו כן, האקסל מצורף כנספח לדו"ח.

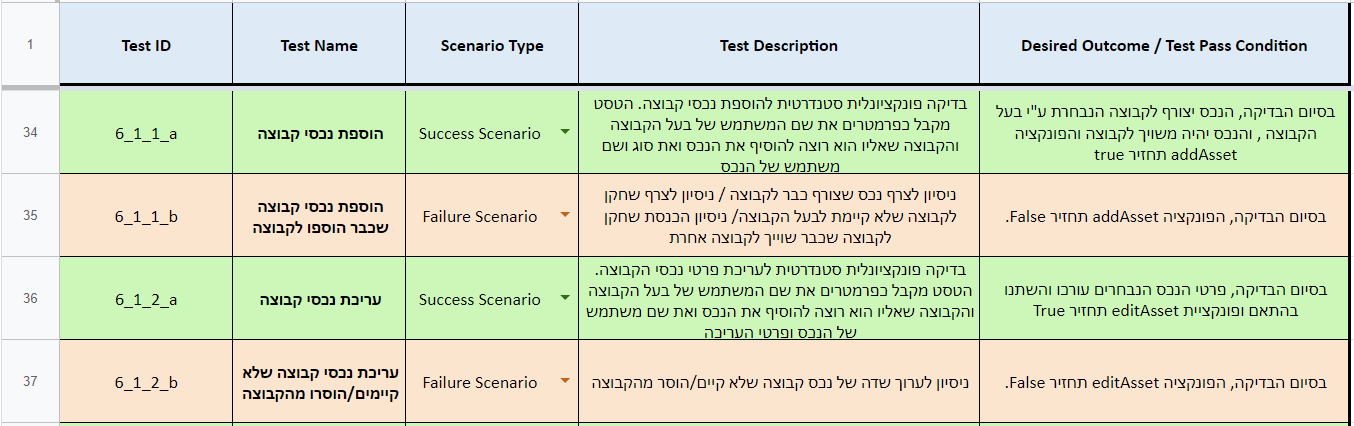
אקסל זה מכיל שני גיליונות:

1. Matrix – גיליון אשר מכיל מטריצה, בה השורות מיוצגות לפי מספר תרחישי השימוש והעמודות לפי Test-ID. כל מבחן קבלה שבו הטסט עובר יסומן ב- ‘V’ במקום המתאים במטריצה, וכאשר הטסט אינו עובר הוא יסומן ב- ‘X’.
2. Tests – פירוט הטסטים. בגיליון Matrix מסומנים בצורה כללית האם הטסטים עברו או לא, וכדי לפרט בצורה רחבה יותר על הבדיקות, יצרנו גיליון נוסף אשר מכיל טבלה של הטסטים עצמם עם ה-Test-ID שלהם, שם הטסט, תרחיש קבלה (חיובי או שלילי), תיאור הטסט ותנאי המעבר:

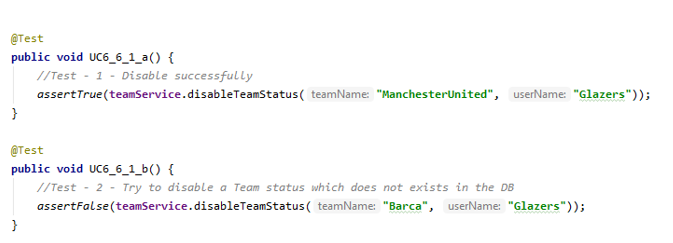


גיליון MATRIX

בקוד עצמו, הבדיקות חולקו לפי ה-Test-ID, בהתאם ל-Test-ID במטריצת העקביות:



גיליון TEST



הקוד