**STR**

**מסמך סיכום תוצאות הבדיקות לאתר**

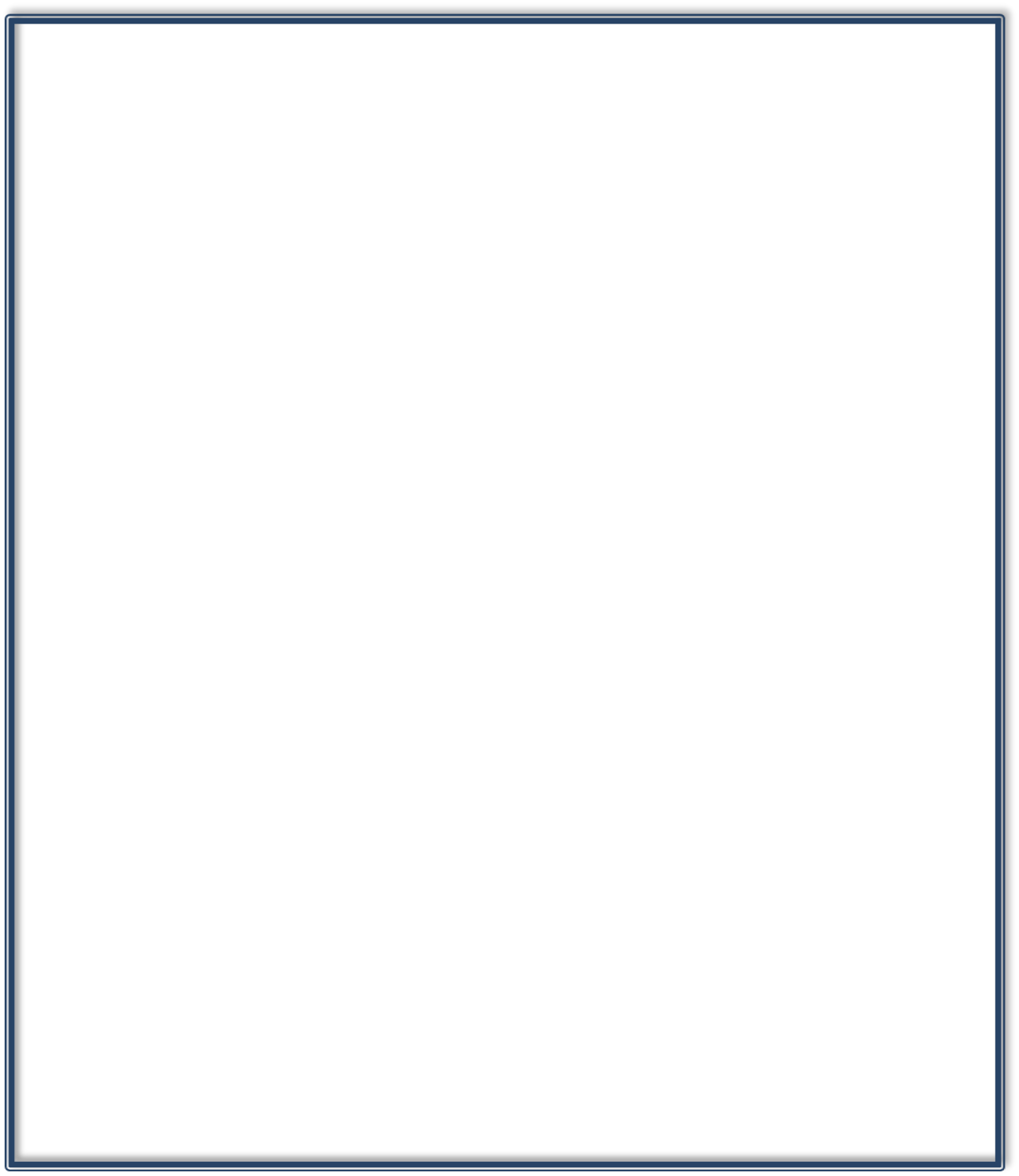
**שופרסל אונליין**



|  |  |
| --- | --- |
| פרויקט: | שופרסל אונליין |
| נכתב ע"י: | צוות QA – ISR - ליאור,אור,מיכל,אסיל |
| תאריך עריכה אחרון: | /01/2023**25** |
| מוגש ע"י: | ראש צוות QA - ISR |

**בקרת תפוצה**

|  |  |  |  |
| --- | --- | --- | --- |
| תאריך | נכתב ע"י | אושר ע"י | תפקיד |
| /01/202325 | צוות QA - ISR | קוסקה רז | מרצה QA |

**תוכן עניינים**

[**1. כללי 3**](#_Toc122885405)

[**1.1. תיאור המערכת 3**](#_Toc122885406)

[**1.2. מסמך תוצאות הבדיקות ויעדיו 3**](#_Toc122885407)

[**1.3. סימוכין 3**](#_Toc122885408)

[**1.4. מונחים ומושגים 4**](#_Toc122885409)

[**2. תהליך הבדיקות 7**](#_Toc122885411)

[**2.1. תיאור ביצוע הבדיקות. 8**](#_Toc122885412)

[**2.2. סיכום ביצוע הבדיקות 8**](#_Toc122885413)

[**2.3. בעיות במהלך ביצוע הבדיקות. 10**](#_Toc122885413)

[**3. תוצאות הבדיקות – הצגת הבאגים 10**](#_Toc122885414)

[**3.1. סיכום באגים פתוחים בסיום התהליך 11**](#_Toc122885416)

[**4. נוהל טיפול בבאגים 12**](#_Toc122885420)

[**5. קריטריונים לאישור המערכת 13**](#_Toc122885421)

[**6. מסקנות והמלצות 14**](#_Toc122885422)

# ***1. כללי***

* 1. **תיאור המערכת**

**שופרסל Online** הינו ממשק אינטרנטי המופעל דרך אתר ואפליקציה סלולרית, שבמסגרתו מתבצעות מכירות קמעונאיות. האתר מנוהל ע"י חברת שופרסל בע"מ המפעילה רשתות מרכולים ובתי מרקחת ואף עוסקת בנדל"ן. הרשת קיימת משנת 1958 כאשר בשנת 2013 השיקה החברה את שופרסל Online. לחברת שופרסל מאות סניפים ברחבי מדינת ישראל, והיא משרתת בתור רשת המזון הוותיקה ביותר בישראל וללא ספק בין הזולות שבהן. שופרסל אונליין נועד לקדם ולייעל את השירות המוענק ללקוחות הרשת ולספק אפשרות לאנשים להזמין מוצרים ישירות לבתיהם באמצעות משלוח וכמו כן קיימת אפשרות איסוף עצמי.  
ניתן להוריד את האפליקציה במכשירי iOS, אנדרואיד ושאר מכשירי הסמארטפון למיניהם ובנוסף ניתן לגלוש ולהגיע אליו דרך אתר האינטרנטי.

**1.2. מסמך תוצאות הבדיקות ויעדיו**מטרת מסך ה-STR היא להציג את תוצאות הבדיקות.  
המסמך מסכים את תהליך הבדיקות, ומתאר את הכשלים שנמצאו, והבעיות עמן נאלצנו להתמודד לאורך התהליך.  
המסמך מכיל:

* הגדרות והסברים למונחים ומושגים.
* תיאור הבדיקות, שבוצעו במערכת ניהול מחסן, וקישורן לדרישות עליהן הן עונות.
* תיאור תהליך פתיחת באג מתחילתו ועד סופו.
* פירוט באגים שנמצאו.
* ניהול סיכונים.
* מסקנות והערכת המערכת.

**1.3. סימוכין**

|  |  |  |
| --- | --- | --- |
| מספר | שם המסמך | תיאור המסמך |
| **1** | **SRS** | **Software Requirements Specification Document – מסמך דרישות תוכנה** |
| **2** | **STP** | **Software Test Plan – מסמך תכנון הבדיקות** |
| **3** | **STD** | **Software Test Design – מסמך עיצוב/תיאור הבדיקות** |

**1.4. מונחים ומושגים**

להלן רשימת מושגים ומונחים, שיעזרו להתמצא במסמכי בדיקות שונים ובחלקם יעשה שימוש לאורך מסמך זה:

|  |  |  |
| --- | --- | --- |
| מס' | שם המושג/המונח: | תיאור המושג/המונח: |
| 1 | STP | מסמך תכנון הבדיקות – Software Test Plan |
| 2 | STD | מסמך עיצוב / תיאור הבדיקות - Software Test Design Description |
| 3 | STR | מסמך תוצאות הבדיקות Software Test Results |
| 4 | SRS | מסמך דרישות לתוכנה עליה נבצע בדיקות |
| 5 | PRD | מסמך דרישות ואפיון לתוכנה עליה נבצע בדיקות |
| 6 | JIRA | תוכנה לניהול פרויקטים, התוכנה מאפשרת מעקב אחר באגים, [פיתוח תוכנה זריז](https://he.wikipedia.org/wiki/%D7%A4%D7%99%D7%AA%D7%95%D7%97_%D7%AA%D7%95%D7%9B%D7%A0%D7%94_%D7%96%D7%A8%D7%99%D7%96), וניהול עבודה, במיוחד לצוותים שעובדים בשיטת ה-Agile ונחשבת לכלי מס' 1 בעולם בתחום |
| 7 | Xray Test Management | כלי לניהול בדיקות ידניות ואוטומטיות המתחבר לתוכנת ניהול פרויקטים, JIRA |
| 8 | TFS | מערכת ניהול כולל של מחזורי הפיתוח. Data Base מרכזי המכיל מידע ומקשר בין שלושת המערכות של Microsoft המשתתפות במחזור הפיתוח – Web Access - כלי הניהול, VS (Visual Studio) – כלי הפיתוח, MTM (Microsoft Test Manager) – כלי הבדיקות |
| 9 | MTM | Microsoft Test Manager – מערכת בדיקות המהווה חלק אינטגרלי מה-Visual Studio, משמשת לכתיבת תסריטי בדיקה ומאפשרת לבחון לעומק את מחזור החיים המשותף של שלבי הפיתוח |
| 10 | QC | Quality Center – כלי בדיקות מבית HP. כלי אינטרנטי (WEB), ריכוזי ובו מסד הנתונים משותף (לבדיקות בלבד), ויכול להיות על שרת מרכזי או בענן. מסד נתוני הכל אינו מסונכרן אוטומטית עם כלי ניהול ופיתוח. הכלי תומך בניהול בדיקה, הרצת בדיקות, דיווח ומעקב אחר אירועים ומאפשר ניתוח והפקת גרפים ודוחות |
| 11 | ALM | Application Lifecycle Management – גרסת QA מורחבת, המיועדת לניהול שלבים שונים במחזור חיי פיתוח תוכנה. כוללת אפשרויות נרחבות לניהול תצורה, הגדרות שונות, השוואה ושמירת היסטוריית שינויים |
| 12 | SDLC | מחזור חיי פיתוח תוכנהSoftware Development Life – Cycle |
| 13 | Traceability | נעקבות – היכולת לזהות פריטים קשורים במסמכים ובקוד (למשל: קישור בין דרישות – לבדיקות – לבאגים) |
| 14 | Web | הפלטפורמה האינטרנטית |
| 15 | Mobile | פלטפורמת המכשיר הנייד |
| 16 | IOS | מערכת הפעלה לניידים של אפל (אייפון) |
| 17 | ANDROID | מערכת הפעלה לניידים של גוגל |
| 18 | HTTP | פרוטוקול להעברת מידע ברשת האינטרנט |
| 19 | Data Base (DB) | מסד נתונים |
| 20 | Test Case (TC) | תרחיש בדיקה – מצומצם – עונה על דרישה פרטנית |
| 21 | User Story (US) | מקרה שימוש / סיפור משתמש – מקרה בדיקה רחב, המתייחס לפעולה עסקית שלמה (טרנזאקציה), מתחילתה ועד סופה |
| 22 | Happy Flow | זרימה תקינה – פעולה עסקית שלמה (טרנזאקציה), מוצלחת – מעבר בכל שלבי התהליך, מתחילתו ועד סופו, ללא תקלות. הרבה בשימוש בבדיקות כיסוי משפטים בקוד, ניתן להשתמש גם בבדיקות שפיות) |
| 23 | Unit/Component Testing | בדיקות יחידה/רכיבים – בדיקות קוד – באחריות צוות הפיתוח |
| 24 | Integrational Testing | בדיקות אינטגרציה – בדיקות שילוב היכולות לבוא לידי ביטוי בבדיקות מבניות, פונקציונאליות ולא פונקציונאליות, וברמות שונות: שילוב בין רכיבים שונים, בין רכיב לפיצ'ר, בין פיצ'רים שונים באותה מערכת, בין מערכות שונות, בין מערכת לסביבה, בין תוכנה וחומרה ועוד |
| 25 | System Testing | בדיקות מערכת – הבודקות את התנהגות המערכת כמכלול. בדיקות בהיקף גדול ,פונקציונאליות ולא פונקציונאליות בעיקר, שמטרתן לוודא כי המערכת עושה את מה שהיא נדרשת לעשות, ולא עושה מה שאינה נדרשת.  הבדיקות מתנהלות תוך איתור, רישום וסיווג כל התקלות שהתגלו במהלך הרצת תרחישי הבדיקה.  לעתים קרובות מדובר בבדיקות המקיפות האחרונות, טרם המסירה ללקוח |
| 26 | Acceptance Testing | בדיקות קבלה – שלב הבדיקות הסופי – לרוב מבוצע ע"י הלקוח / המשתמש – בדיקות מוכנות המערכת.  מטרתן לוודא כי המערכת פועלת בהתאם לדרישות, לבסס בטחון במערכת, ולוודא כי היא מוכנה לעלייה לאוויר |
| 27 | Functional Testing | בדיקות פונקציונאליות – בודקות **מה** המערכת עושה |
| 28 | Non-Functional Testing | בדיקות לא פונקציונאליות – בודקות **איך** המערכת עושה |
| 29 | (N) Not Run | תרחיש בדיקה שלילי (נגטיבי) – בדיקות המכסות מקרים בהם מכניסים קלט שגוי / אסור למערכת, ומצפים שהמערכת תישאר יציבה ותמשיך לעבוד בליווי הודעת שגיאה |
| 30 | Sanity / Smoke Tests | בדיקות שפיות / עשן – אחוז הבדיקות הקריטיות והחשובות ביותר במוצר, המהוות את התפקוד העיקרי של המערכת, "תרחישי סף" שבמידה ונכשלים אין טעם להמשיך לבדוק ויש להחזיר לפיתוח |
| 31 | Confirmation Tests | בדיקות אימות – לאחר ביצוע תיקון במערכת (תיקון באג). בדיקות חוזרות של אותם חלקים שנמצאים תקולים במערכת לשים וידוא כי אכן תוקן הכשל כשנדרש |
| 32 | Regression Tests | בדיקות רגרסיה (נסיגה) – לאחר ביצוע שינוי במערכת.  בדיקות חלקים במערכת שתפקדו טרם השינוי ולא שונו ,אך עלולים להיות מושפעים מהשינוי, בכדי לוודא כי לא נפגעו בעקבותיו.  לרוב מדובר בבדיקות חוזרות-שבוצעו בהצלחה בעבר |
| 33 | Monkey / Exploratory Testing | בדיקות "שיטוט חופשי"- המתבצעות ללא תסריטי בדיקה |
| 34 | Passed (Test) | הבדיקה הורצה – ועברה בהצלחה |
| 35 | Failed (Test) | הבדיקה הורצה – ונכשלה |
| 36 | Blocked (Test) | בדיקה בסטאטוס "חסום "– לא ניתן להריץ את הבדיקה ,עקב סיבות שונות כגון: חוסר ידע, טכנולוגיה מתאימה, או מכשור ועוד |
| 37 | Not run (Test) | בדיקה שאינה מבוצעת מסיבות שונות, על אף שנכתבה |
| 38 | Code Freeze (CF) | ביטוי להפסקת עבודה בפיתוח המערכת – עד לאיטרציה הבאה.  ברגע שהוכרז על Code Freeze (לרוב בשלושת הימים האחרונים של האיטרציה) ,לא מתבצעים שינויים ולא נוצרות תתי גרסאות חדשות לאותה איטרציה, על מנת שצוות ה- QA יבדוק את המערכת לאחר הפיתוח (צוות הפיתוח אמון בזמן זה על תיקון באגים שהתגלו ) |
| 39 | Feature Freeze (FF) | ביטוי להפסקה בפיתוח פיצ'רים – עד לאיטרציה הבאה.  עד לנקודת הזמן בה מוכרז על Feature Freeze, ניתן לבצע שינויים מינוריים – בינוניים בפיצ'רים, בתיאום עם מנהל הפיתוח, מעבר לנקודת זמן זו – זה נשאר לאיטרציה הבאה |
| 40 | Hardening (Iteration) | להקשיח את המערכת – איטרציית עבודה בה לא מתקבלות דרישות חדשות – שמורה לבדיקות רגרסיה ולתיקון באגים – בצוות הפיתוח |
| 41 | GUI / UI | וויזואליות / נראות המערכת .  כל הקשור בממשק המשתמש כגון: גודל וצבע גופן, צבעי רקע ,שדות, מסגרות תוכן, תמונות, כפתורים וכדומה |
| 42 | UX | חווית המשתמש - User Experience  כשמה כן היא: חווית משתמש הקצה בזמן פעילותו במערכת.  נוחות השימוש במערכת וחוויית המשתמש, משיקה ומושפעת מממשק המשתמש- UI |
| 43 | טרנזאקציה | רצף פעולות, המהוות תהליך עסקי שלם, מתחילתו ועד סופו |
| 44 | אתר אינטרנט | אוסף של דפי אינטרנט המקושרים ביניהם, ולרוב שותפים לתחום מסוים ,הניתנים לגישה דרך רשת האינטרנט |
| 45 | אפליקציה | יישומון – יישום מחשב ,המיועד לשימוש בטלפונים  חכמים, מחשבי לוח ומכשירים ניידים מסוגים אחרים. |
| 46 | דפדפנים | מכונת גלישה בדפי אינטרנט |
| 47 | Bug | תוצאה שגויה / לא צפויה של המערכת |
| 48 | Bug Fixes | תיקוני באגים – תיקון תוצאה שגויה / לא צפויה של המערכת |
| 49 | Critical Bug | באג קריטי.  המונע מהלקוח להשתמש במערכת, וגורם לקריסות.  באג שחובה לטפל בו, אחרת לא ניתן לשחרר גרסה / מערכת |
| 50 | High Level Bug | באג חשוב.  ברמת חומרה גבוהה ,המפריע לפעילות תקינה של המערכת, אך אינו גורם לקריסה , וניתן לעקיפה. חשוב לטפל בבאג מסוג זה, ולא מומלץ לשחרר גרסה/ מערכת, המכילה באגים High Level |
| 51 | Medium Level Bug | באג ברמת חומרה בינונית.  מאפשר למשתמש להמשיך לעבוד.  לעתים יכול לגרום לעיכובים בשחרור גרסה / מערכת |
| 52 | Low Level Bug | באג ברמת חומרה נמוכה.  לרוב קשור באינטרפייס המשתמש (נראות / ממשק), ואינו פוגע בפונקציונאליות המערכת.  לרוב יטופלו בעדיפות אחרונה, ולא יעכבו שחרור גרסה / מערכת |
| 53 | Scrum Master | מנהל איטרציה – אדם שתפקידו לדאוג לפתרון הבעיות שעלו באיטרציה |
| 54 | Sprint / Iteration | ספרינט / איטרציה – מחזור פיתוח במודל האג'ילי – אורך לרוב בין שבועיים לארבעה שבועות |
| 55 | Agile - Model | מודל פיתוח זריז – מודל מחזורי מבוסס סבבים.  מותאם לפיתוח חלקים קטנים בסבבים מהירים ,ספרינטים / (איטרציות) של עד כחודש. בכל שלב, תהליך הפיתוח מקביל לתהליך הבדיקות, וקיים עירוב מתמיד של הלקוח, המאפשר מענה לשינויים בדרישות לאורך התהליך |
| 56 | Waterfall – WF Model | מודל פיתוח מפל המים – מודל לינארי (קווי). מחזור פיתוח בודד, שלב הבדיקות מגיע לאחר שלב אחד בפיתוח ולא במקביל. לאחר מעבר לשלב הבא, לא חוזרים לשלב הקודם. המודל שם דגש על עיצוב מוקדם של התוכנה, וניסיון לצפות שינויים עתידיים. מודל ארוך ומתועד שאינו מתאים לסבבי פיתוח קצרים. אין עירוב של הלקוח לאורך תהליך הפיתוח |
| 57 | Dev (Environment) | סביבת העבודה של הפיתוח.  כמעט לא מבוקרת, אין דיווח שינויים ("אין חוקים"), בקרה עיקרית – שהסביבה תעבוד |
| 58 | QA (Testing Environment) | סביבת העבודה של הבדיקות.  סביבה מבוקרת, דיווח שינויים, בסיום עבודה, מוחזרת למצבה ההתחלתי. משתדלת להיות קרובה ככל הניתן לסביבת הלקוח |
| 59 | STG (Environment) | סטייג'ינג Stagin - סביבה יקרה מאוד – מדמה 1:1 את סביבת הלקוח.  מאפשרת לדמות את המצב הנוכחי, ולבדוק כיצד יושפע משינויים מסוימים, טרם העליה לאוויר |
| 60 | PROD (Environment) | פרודקשן – production (סביבת הלקוח – המערכת באוויר ) |

# ***תהליך הבדיקות***

**בדיקות Sanity (שפיות)**הבדיקות הראשונות שמבצעים, בדיקות בסיסיות וקריטיות, המאפשרות לזהות במהירות וביעילות, אם הפונקציונליות הבסיסית / העיקרית של המוצר, פועלת כנדרש.  
בדיקות Sanity מהוות "תרחישי סף", במידה וקיימת בעיה באחד השלבים, אין טעם להמשיך לבדוק את המוצר, חייבים קודם לתקן את הבאגים הקיימים, ויש להחזיר לפיתוח.

**בדיקות System (מערכת)**בדיקות המערכת – בודקות את התנהגות המערכת כמכלול בדיקות בהיקף גדול, פונקציונאליות ולא פונקציונאליות בעיקר, שמטרתן לוודא כי המערכת עושה את מה שהיא נדרשת לעשות, ולא עושה מה שאינה נדרשת .  
הבדיקות תכלולנה בין היתר: בדיקות מסדי נתונים, קשרי גומלין בין תהליכים, קשרים עם מערכות חיצוניות, בדיקות ביצועים, בדיקות עומסים, שרידות, שימושיות, בדיקות אבטחת מידע, ובדיקות התאוששות ממצבי כשל.  
בדיקות המערכת מתנהלות תוך איתור, רישום וסיווג כל התקלות שהתגלו במהלך הרצת תרחישי הבדיקה.  
לעתים קרובות מדובר בבדיקות המקיפות האחרונות, טרם מסירת המערכת ללקוח.

**בדיקות Regression (רגרסיה)**לאחר ביצוע שינוי כלשהו במערכת, בין אם נובע מעדכון גרסה, או מתיקון באג, מלבד בדיקות אימות, שנועדו לווידוא תיקון הבאג, יש צורך בבדיקות רגרסיה.  
בדיקות החלקים במערכת שתפקדו טרם השינוי, ולא שונו, אך עלולים להיות מושפעים מהשינוי, במטרה לוודא כי לא נפגעו בעקבותיו.  
לרוב מדובר בבדיקות חוזרות – שבוצעו בהצלחה בעבר.

**Monkey Testing /Exploratory (שיטוט חופשי)**טרם מסירת המערכת, רצוי לבצע בדיקות "שיטוט חופשי" - המתבצעות ללא תסריטי בדיקה, ומדמות את פעולת המשתמש, ל"חיזוק אחרון " ברמת הביטחון במערכת ובתקינותה.

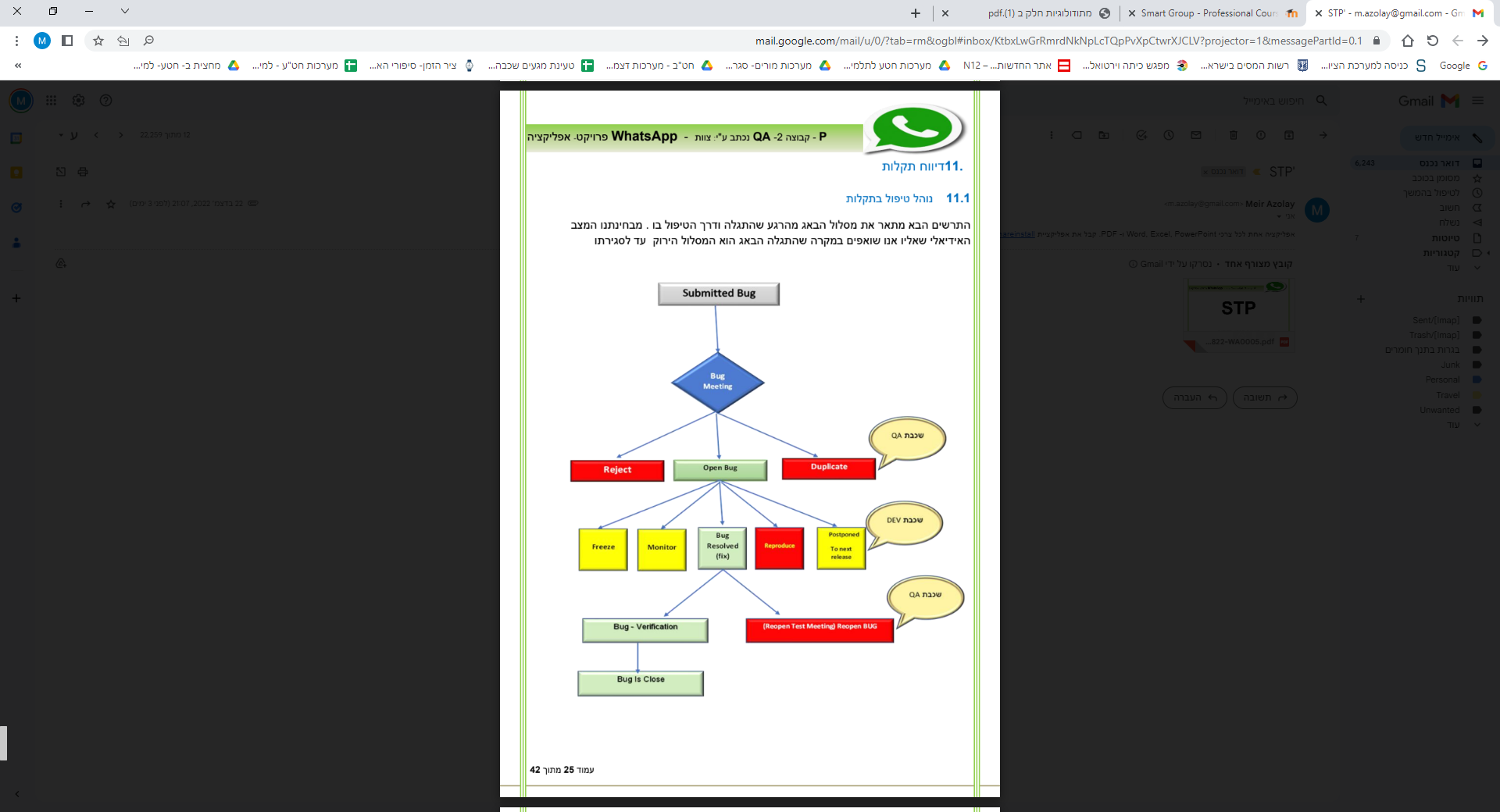
**2.1. תיאור ביצוע הבדיקות**  
הבדיקות בוצעו ב-2 סבבים (כל סבב ארך שבוע).  
כל סבב התמקד בגרסת מערכת אחת: סבב 1 – גרסה 1.0.0 , סבב 2 – גרסה 1.0.1.  
בכל סבב דרישות המערכת נבדקו עבור כל אחת מ-2 מערכות ההפעלה התומכות את המערכת:  
מחשב PC (Windows 10 Pro, Windows 11 Pro) ומובייל (iOS+Android) וב-3 סוגי דפדפנים: Google Chrome ו-Mozilla Firefox ו-Opera בהתאם לאיפיון.  
  
הקונפיגורציות נבחרו בתיאום עם האחראי, בהתאם לדרישות, ובהתחשב בלוחות הזמנים והאמצעים הטכניים העומדים לרשותינו.  
הבדיקות תוכננו לבחינת: ממשק משתמש, תקינות נתונים, תאימות, פונקציונליות ותהליכיות,   
לצד אבטחת המידע, עומסים וביצועי המערכת.  
(על סט הבדיקות הטכניות אמון צוות הפיתוח – בדיקות אלו אינן מתוארות במסמך זה).  
 **הסבב הראשון כלל:** בדיקות “Site Design” טכניות (באחריות צוות הפיתוח), בדיקות שפיות – “Happy Flow” למספר פעולות עסקיות נבחרות, המהוות את הפונקציות העיקריות הבסיסיות של המערכת (בדיקות אלו כללו גם בדיקות התקנה והסרה), ונבדקו 8 דרישות.  
  
**הסבב השני כלל:** בדיקות שפיות, אימות ורגרסיה מלאה עבור בדיקות הסבב הראשון, ובדיקות "שיטוט חופשי" (Exploratory), המדמות את פעילות משתמש הקצה, לשם הבטחת איכות המערכת.  
  
  
**2.2. סיכום ביצוע הבדיקות**  
  
סה"כ תוכננו 1284 טסטים, מתוכם בוצעו 1284 טסטים, בשתי הקונפיגורציות השונות, בשני הסבבים. כל הטסטים המתוכננים בוצעו.   
  
האחוזים הבאים מראים רק את הטסטים של האיטרציה השניה – גרסה 1.0.1 הרלוונטית (642 טסטים תוכננו באיטרציה זו סה"כ – Full Regression):  
  
 600 טסטים, המהווים 93.5% - עברו בהצלחה.  
 42 טסטים, המהווים 6.5% - נכשלו.  
  
  
  
  
**סך הטסטים שעברו/נכשלו באיטרציה הראשונה והשנייה**  
  
  
  
**2.3. בעיות במהלך ביצוע הבדיקות**  
 נפילת שרתים בגלל עומס על המערכת למשך יום עבודה שלם.  
 בדיקות לא הורצו במערכת בגלל בעיות טכניות, מה שגרם לעיכוב של יום עבודה שלם אך הגרסה החדשה נמסרה ללא עיכוב כמתוכנן.

# ***תוצאות הבדיקות – הצגת באגים*** להלן תרשימים המתארים את התפלגות הבאגים, ע"פ חומרה, לאורך 2 הסבבים:

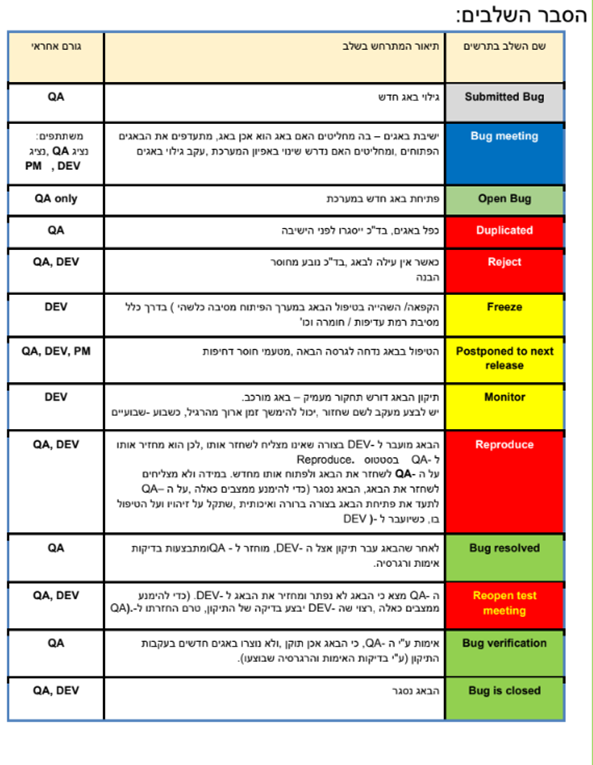
## 3.1. סיכום באגים פתוחים בסיום התהליך בסבב הבדיקות הראשון, נפתחו 42 באגים בכל הרמות: נמוך – קריטי, כש1 מתוכם קריטיים, תוקנו במסגרת Bug Fixes. בסבב הבדיקות השני, נפתחו 41 באגים ברמות: נמוך – גבוה, כאשר הבאג הקריטי נפתר. לסיכום: בסיום תהליך בדיקות זה, נותרו 41 באגים פתוחים: 25 ברמה נמוכה, 12 ברמה בינונית, 4 ברמה גבוהה, 0 ברמה קריטית.

## 

***4. נוהל טיפול בבאגים***



## 4.1. הסבר השלבים



***5. קריטריונים לאישור המערכת***

**טסטים:**

|  |  |  |
| --- | --- | --- |
| **בפועל** | **יעד** | **קריטריון** |
| 97% | 100% | ) שעברו בהצלחהSanity% בדיקות שפיות ( |
| 100% | 90% | % בדיקות שבוצעו מתוך בדיקות שתוכננו |
| 94% | 87% | % בדיקות שעברו בהצלחה מתוך בדיקות שבוצעו |
| 1% | 3% | (Blocked)% בדיקות בסטטוס "חסום" |
| 0% | 1% | )Not Run% בדיקות שתוכננו ולא בוצעו (לא מסיבה חסימה -  כל הבדיקות שלא הורצו היו באיטרציה הראשונה – גרסה 1.0.0 שהוחזרה לפיתוח |

**באגים:**

|  |  |  |
| --- | --- | --- |
| **בפועל** | **יעד** | **קריטריון** |
| 0 | 0 | Criticalבאגים פתוחים ברמת חומרה |
| 4 | 0 | Highבאגים פתוחים ברמת חומרה |
| 2% | 5% | (מקסימום % טסטים שנכשלו)Mediumבאגים פתוחים ברמת חומרה |
| 4% | 10% | (מקסימום % טסטים שנכשלו)Lowבאגים פתוחים ברמת חומרה |

***6. מסקנות והמלצות***לסיכום, כמתואר במסמך זה, המערכת עברה את מירב הבדיקות בהצלחה, ועמדה בקריטריונים וביעדים שנקבעו בדרישות, לכן אנו רואים כי ניתן לשחרר את גרסה 1.0.1 ולהעלותה לאוויר.  
  
חשוב לציין כי כל התקלות שאובחנו במהלך הבדיקות, טופלו ונסגרו.  
לאחר בדיקת גרסה 1.0.1 נמצא כי ב-94% מהבדיקות שבוצעו עברו בהצלחה ו-6% נכשלו, ורובם ברמת חומרה נמוכה ובינונית, ורק 1% מהם גבוהה, ולא נמצאו בכלל בדיקות שנכשלו ברמה קריטית. מכאן, מצאנו שהמערכת מהווה כלי יעיל ונוח למשתמשים, עובדי ארגונים, לנהל את ,ונבדקה במקבץ דפדפנים.Mobile,Webהמידע האישי שלהם. המערכת יציבה בסביבת ה-