STR

מסמך סיכום תוצאות הבדיקות למערכת ניהול מחסן

|  |  |
| --- | --- |
| System Testing | פרוייקט |
| Omri Eden | נכתב ע"י |
| 20/09/2022 | תאריך עריכה אחרון |
| Omri Eden- QA Manager | מוגש ע"י |

בקרת תפוצה

|  |  |  |  |
| --- | --- | --- | --- |
| תפקיד | אושר ע"י | נכתב ע"י | תאריך |
| PM | חוטאיפה זועבי | Omri Eden | 09/22 |
|  |  |  |  |
|  |  |  |  |

תוכן עניינים

1. כללי...............................................................................................3
   1. תיאור המערכת...................................................................3
   2. תוצאות הבדיקות מהו STR...................................................3
   3. סימוכין..............................................................................4
   4. מונחים ומושגים...................................................................4
2. תהליך הבדיקות..................................................................................7

2.1 תיאור ביצוע הבדיקות...........................................................8

2.2 סיכום ביצוע הבדיקות...........................................................9

2.3 בעיות במהלך ביצוע הבדיקות...............................................11

1. תוצאות הבדיקות- הצגת באגים............................................................12

3.1 סיכום באגים פתוחים בסיום התהליך......................................13

1. נוהל טיפול בבאגים............................................................................14
2. קריטריונים לאישור המערכת...............................................................15
3. מסקנות והמלצות...............................................................................16
4. **כללי**
   1. **תאור המערכת**

**מערכת זו מיועדת לניהול מחסן ומכירות של חברה המספקת ציוד לעבודות יד, שיפוצים ובנייה. המערכת באה להחליף מערכת קיימת שכבר איננה מתאימה לצורכי החברה (המערכת הישנה כללה רק מאגר פריטים ומאגר לקוחות שעודכנו ידנית, מאגרים אלו הועברו למערכת החדשה בתהליך של הסבת נתונים). המערכת נתמכת גם באתר אינטרנט ייעודי שאליו ניתן להתחבר מחוץ לחברה, אתר זה מיועד לעבודה על הגרסאות האחרונות של Chrome ו- Firefox אך אינו נתמך בגרסאות .Explorer**

**1.2.** **תוצאות הבדיקות ומהו מסמך STR**

**מטרת מסמך הSTR היא להציג את תוצאות הבדיקות. המסמך מסכם את תהליך הבדיקות, ומתאר את הכשלים שנמצאו, והבעיות עמן נאלצנו להתמודד לאורך התהליך.**

**המסמך מכיל:**

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

**1.3. סימוכין**

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

**1.4. מונחים ומושגים**

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

|  |  |
| --- | --- |
| **תיאור המושג / המונח** | **מושג / מונח** |
| **מסמך דרישות ואיפיון מוצר – Product Requirements Document** | **PRD** |
| * **Software Requirements Specification Document**   **מסמך דרישות ואפיון תוכנה** | **SRS** |
| **מסמך תכנון הבדיקות – Software Test Plan** | **STP** |
| **מסמך עיצוב / תיאור – Software Test Design / Description**  **הבדיקות** | **STD** |
| **מסמך תוצאות הבדיקות – Software Test Results** | **STR** |
| **Team Foundation Server – מערכת לניהול כולל של מחזור הפיתוח.**  **Data Base מרכזי המכיל מידע ומקשר בין שלושת המערכות של**  **Microsoft המשתתפות במחזור הפיתוח.** | **TFS** |
| **Microsoft Test Manager – כלי בדיקות מבית Microsoft, העובד בסנכרון עם כלי הניהול וכלי הפיתוח.**  **ניהול בדיקות באמצעות כלי זה לצד הכלים הנ"ל, מפשט את הקישור והנעקבות בין דרישות, טסטים ובאגים, והופך את תהליך הפיתוח לאינטגרטיבי** | **MTM** |
| **Quality Center – כלי בדיקות מבית HP. זהו כלי אינטרנטי בו מסד הנתונים משותף, ויכול להיות על שרת מרכזי או בענן.**  **מסד נתוני הכלי אינו מסונכרן אוטומטית עם כלי ניהול ופיתוח.**  **הכלי תומך בניהול בדיקה, הרצת בדיקות, ודיווח ומעקב אחר ארועים.**  **כמו כן, מאפשר ניתוח והפקת גרפים ודו"חו ת** | **QC** |
| **Application Lifecycle Management – גרסת QC מורחבת המיועדת לניהול השלבים השונים במחזור חיי פיתוח התוכנה.**  **כוללת אפשרויות נרחבות לניהול תצורה, הגדרות שונות, השוואה ושמירת הסטוריית שינויים .** | **ALM** |
| **מחזור חיי פיתוח תוכנה – Software Development Life Cycle** | **SDLC** |
| **נעקבות – היכולת לזהות פריטים קשורים במסמכים ובקוד )למשל :**  **קישור בין דרישות – לבדיקות – לבאגים(** | **Traceability** |
| **הפלטפורמה האינטרנטית** | **Web** |
| **פלטפורמת המכשיר הנייד** | **Mobile** |
| **מערכת הפעלה לניידים של אפל** | **IOS** |
| **מערכת הפעלה לניידים של גוגל** | **ANDROID** |
| **פרוטוקול להעברת מידע ברשת האינטרנט** | **HTTP** |
| **מסד נתונים** | **Data Base (DB)** |
| **תרחיש בדיקה – מצומצם – עונה על דרישה פרטנית** | **Test Case (TC)** |
| **מקרה שימוש / סיפור משתמש – מקרה בדיקה רחב, המתייחס לפעולה עסקית שלמה מתחילתה ועד סופה .** | **User Story (US)** |
| **זרימה תקינה – פעולה עסקית שלמה מוצלחת – מעבר בכל שלבי התהליך, מתחילתו ועד סופו, ללא תקלות. הרבה בשימוש בבדיקות כיסוי משפטים בקוד, ניתן להשתמש גם בבדיקות שפיות** | **Happy Flow** |
| **בדיקות יחידה / רכיבים – בדיקות קוד – באחריות צוות הפיתוח** | **Unit / Component 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** | **Integrational 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.** | **System 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** | **Acceptance tests** |
| **בדיקות פונקציונאליות – בודקות מה המערכת עושה** | **Functional testing** |
| **בדיקות לא פונקציונאליות – בודקות איך המערכת עושה** | **Non-Functional testing** |
| **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** | **Sanity / Smoke tests** |
| **בדיקות נסיגה- לרוב לאחר תיקון באג במערכת. בדיקות חלקים במערכת שתפקדו טרם השינוי ולא שונו ,אך עלולים להיות מושפעים מהשינוי, בכדי לוודא כי לא נפגעו בעקבותיו.** | **Regression tests** |
| **בדיקות "שיטוט חופשי "- המתבצעות ללא תסריטי בדיקה** | **Monkey testing** |
| **הבדיקה הורצה – ועברה בהצלחה** | **Passed (Test)** |
| **הבדיקה הורצה – ונכשלה** | **Failed (Test)** |
| **בדיקה בסטאטוס "חסום "– לא ניתן להריץ את הבדיקה ,עקב סיבות שונות .** | **Blocked (Test)** |
| **בדיקה שאינה מבוצעת מסיבות שונות, על אף שנכתבה** | **Not run (Test)** |
| **ביטוי להפסקת עבודה בפיתוח המערכת – עד לסבב הבא. ברגע שהוכרז Code Freeze לא מתבצעים שינויים ולא נוצרות תתי גרסאות חדשות לאותו סבב, על מנת שצוות ה QA יבדוק את המערכת לאחר הפיתוח.** | **Code Freeze (CF)** |
| **ביטוי להפסקה בפיתוח פיצ'רים – עד לסבב הבאה.** | **Feature Freeze (FF)** |

|  |  |  |  |
| --- | --- | --- | --- |
| **להקשיח את המערכת – איטרציית עבודה בה לא מתקבלות דרישות חדשות – שמורה לבדיקות רגרסיה ולתיקון באגים .** | | **Hardening (Iteration)** | |
| **וויזואליות / נראות המערכת.**  **כל הקשור בממשק המשתמש כגון: גודל וצבע גופן, צבעי רקע, שדות, מסגרות תוכן, תמונות, כפתורים וכדומה** | | **GUI / UI** | |
| **חוויית משתמש.**  **כשמה כן היא: חווית משתמש הקצה בזמן פעילותו במערכת.**  **נוחות השימוש במערכת וחוויית המשתמש** | **UX** | |
| **רצף פעולות, המהוות תהליך עסקי שלם, מתחילתו ועד סופו** | | **טרנזאקציה** | |
| **יישומון – יישום מחשב, המיועד לשימוש בטלפונים חכמים, מחשבי לוח טאבלטים ועוד.** | | **אפליקציה** | |
| **תוצאה שגויה / לא צפויה של המערכת** | | **Bug** | |
| **תיקוני באגים – תיקון תוצאה שגויה / לא צפויה של המערכת** | | **Bug Fixes** | |
| **באג קריטי.**  **המונע מהלקוח להשתמש במערכת, וגורם לקריסות.**  **באג שחובה לטפל בו, אחרת לא ניתן לשחרר גרסה / מערכת** | | **Critical 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 חדשה עם מערכת המכילה באגים High Level** | | **High Level Bug** | |
| **באג ברמת חומרה בינונית.**  **מאפשר למשתמש להמשיך לעבוד.**  **לעתים יכול לגרום לעיכובים בשחרור גרסה / מערכת** | | **Medium Bug** | |
| **באג ברמת חומרה נמוכה.**  **לרוב קשור באינטרפייס המשתמש נראות / ממשק ואינו פוגע בפונקציונאליות המערכת.**  **לרוב יטופלו בעדיפות אחרונה, ולא יעכבו שחרור גרסה / מערכת** | | **Low Level Bug** | |
| **מנהל איטרציה – אדם שתפקידו לדאוג לפתרון הבעיות שעלו באיטרציה** | | **Scrum Master** | |
| **ספרינט/ איטרציה – מחזור פיתוח במודל האג'ילי – אורך לרוב בין שבועיים לארבעה שבועו ת** | | **Sprint / Iteration** | |
| **מודל פיתוח זריז – מודל מחזורי מבוסס סבבים.**  **מותאם לפיתוח חלקים קטנים בסבבים מהירים של עד כחודש**  **בכל שלב, תהליך הפיתוח מקביל לתהליך הבדיקות, וקיים עירוב מתמיד של הלקוח, המאפשר מענה לשינויים בדרישות לאורך התהליך** | | **Agile (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 קודם.**  **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 הפיתו ח** | | **Waterfall (WF – Model)** | |
| **סביבת העבודה של הפיתוח.**  **כמעט לא מבוקרת, אין דיווח שינויים "אין חוקים".** | | **DEV (Environment)** | |
| **סביבת העבודה של הבדיקות.**  **סביבה מבוקרת, דיווח שינויים, בסיום עבודה, מוחזרת למצבה ההתחלתי.**  **משתדלת להיות קרובה ככל הניתן לסביבת הלקוח** | | **QA / Testing (Environment)** | |
| **סטייג'ינג Staging– סביבה יקרה מאוד – מדמה 1:1 את סביבת הלקוח.**  **מאפשרת לדמות את המצב הנוכחי, ולבדוק כיצד יושפע משינויים מסוימים, טרם העליה לאוויר** | | **STG (Environment)** | |
| **Production – סביבת הלקוח – המערכת באוויר** | | **PROD (Environment)** | |

**2. תהליך הבדיקות**

**בדיקות Sanity (שפיות)**

**הבדיקות הראשונות שמבצעים – מבוצע ע"י הבודקים . בדיקות בסיסיות וקריטיות, המאפשרות לזהות במהירות וביעילות, אם הפונקציונ אליות 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.**

**בדיקת Regression (נסיגה)**

**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**

**טרם מסירת המערכת, רצוי לבצע בדיקות "שיטוט חופשי" – המתבצעות ללא תסריטי בדיקה, ומדמות את פעולת המשתמש, ל" חיזוק אחרון" ברמת הבטחון במערכת ובתקינותה.**

**2.1 תיאור ביצוע הבדיקות**

**הבדיקות בוצעו ב2 שלבים, בשלב הראשוני – בדקנו את מרבית המודולים המשתתפים במערכת ניהול המחסן.**

**לאחר מכן בשלב השני- ביצענו בדיקות הסבת נתונים אל המערכת החדשה .**

**בדקנו שאכן ההסבה אעברה בצורה נכונה מבחינה כמותית ומבחינה איכותית כלומר שכל הטקסטים עברו כמו שצריך ( לא הפוכים או שלא היה חסר טקסט) ובדקנו שהכמות פריטים הועברה כמו שצריך למערכת החדשה.**

**הבדיקות שביצענו היו :**

**1- בדיקת שפיות- לוודא שהמערכת החדשה אכן עולה כמו שצריך על מערכות ההפעלה ועובדת כמו שצריך.**

**2- ביצענו בדיקות פונקציונאליות מקיפות על מודלי המערכת.**

**3- בדיקות ניידות ותאימות לגרסאות הדפדפנים השונים וממכשיר אנדרואיד.**

**4- בדיקות GUI - בהיבט של חווית משתמש בעיקר.**

**5- בדיקות אבטחה (בהקשר של הזדהות לקוח המתחבר דרך אפליקצייה).**

**6- בדיקות הסבת נתונים של המערכת הישנה אל החדשה.**

**כל ניהול הבדיקות התבצע על ידי מערכת MTM כדי שכל אחד מצוות הפיתוח יוכל לעדכן סטטאוס בהתאם .**

**2.2 סיכום ביצוע הבדיקות**

**סה"כ תוכננו 400 טסטים, מתוכם בוצעו 315 טסטים.**

**בסבב הראשון- תוכננו 300 טסטים ובסבב השני- 100 טסטים.**

**בסבב הראשון- לא בוצעו 85 בדיקות , בשל דרישה של מנהל הפרוייקט לבצע מספר בדיקות רגרסיה נוספות במקום בדיקות אחרות שתוכננו. בנוסף בשל עזיבה של עובד, התעכבו הבדיקות עד לסיום חפיפה של עובד חדש.**

**בסבב השני – בוצעו כל בדיקות הסבת הנתונים.**

**נפרט את סבב הבדיקות הראשון:**

* **תוכננו 300 טסטים**
* **מתוכם בוצעו 215 טסטים שזה 71% מסך הטסטים שתוכננו לסבב הראשון**
* **200 טסטים- עברו בהצלחה המהווים 93% מסך הטסטים שבוצעו בסבב הראשון.**
* **15 טסטים- נכשלו המהווים 7% מסך הטסטים שבוצעו בסבב הראשון**

**השוואה בין כמות הטסטים לכמות הבאגים בסבב הראשון:**

**כעת נפרט את סבב הבדיקות השני:**

* **תוכננו 100 טסטים**
* **מתוכם בוצעו 100 טסטים**
* **95 טסטים- עברו בהצלחה המהווים 95% מסך הטסטים שבוצעו בסבב השני.**
* **5 טסטים- נכשלו המהווים 5% מסך הטסטים שבוצעו בסבב השני.**

**השוואה בין כמות הטסטים לכמות הבאגים בסבב השני**

**2.3. בעיות במהלך ביצוע הבדיקות**

* **באגים מרובים בסבב הראשון גרמו לפרוייקט להתעכב בשל בדיקות רגרסיה מרובות, מנהל הפרוייקט החליט לוותר על מספר בדיקות בשביל לעמוד כמה שיותר בלוחות הזמנים**
* **בודק תוכנה מהצוות החליט להתפטר באמצע פרוייקט הבדיקות, מה שגרם לעיכוב נוסף בשל חלוקה מרובה על שאר העובדים ובנוסף לקח זמן למציאת עובד חדש וחפיפתו.**

**3. תוצאות הבדיקות- הצגת באגים**

**להלן תרשימים המתארים את התפלגות הבאגים, עפ"י חומרה, לאורך 2 סבבים:**

|  |  |  |  |
| --- | --- | --- | --- |
| **כמות הבאגים עפ"י חומרה וסבבי הבדיקות** | | | |
| **סה"כ** | **סבב 2** | **סבב 1** | **חומרת הבאג** |
| **6** | **4** | **2** | **Low** |
| **8** | **1** | **7** | **Medium** |
| **6** | **0** | **6** | **High** |
| **0** | **0** | **0** | **Critical** |
| **20** | **5** | **15** | **סה"כ** |

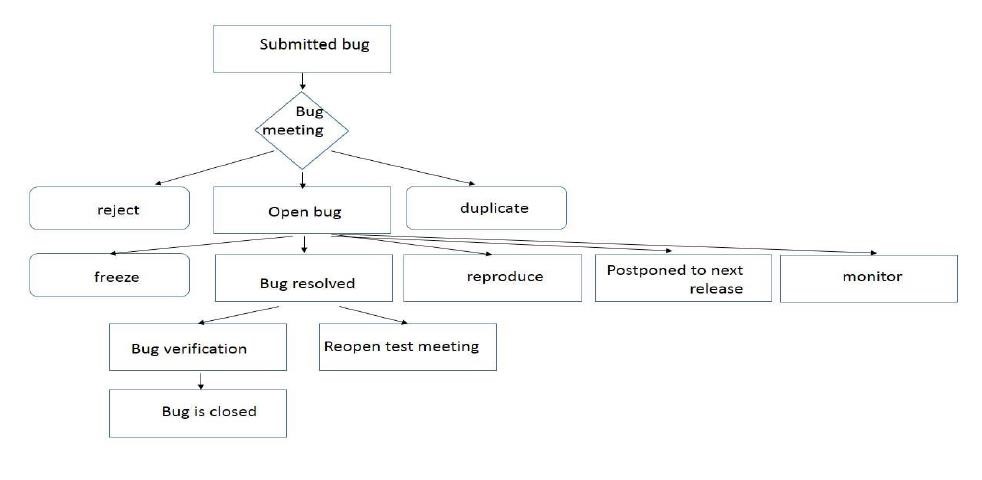
**3.1. סיכום באגים פתוחים בסיום התהליך**

* **בסבב הבדיקות ה-1 - נפתחו 15 באגים ב3 רמות שונות, 2 הבאגים ברמת Low לא תוקנו, הבאגים ברמת Medium ו High Levelתוקנו.**
* **בסבב הבדיקות ה-2 - נפתחו 5 באגים, הבאג ברמת Medium תוקן. 4 הבאגים ברמת Low יתוקנו בהמשך עם תחזוקת המערכת.**

**לסיכום:**

**בתום תהליך הבדיקות נותרו 5 באגים פתוחים ברמת Low Level הוחלט שיתוקנו בהמשך עם תחזוקת המערכת. כולן בעיות GUI.**

**4. נוהל טיפול בבאגים**



**הסבר השלבים:**

|  |  |  |
| --- | --- | --- |
| **גורם אחראי** | **תיאור המתרחש בשלב** | **שם השלב בתרשים** |
| **QA** | **גילוי באג חדש** | **Submitted bug** |
| **משתתפים:**  **נציג QA**  **נציג DEV**  **PM** | **ישיבת באגים – בה מחליטים האם באג הוא אכן באג, מתעדפים את הבאגים הפתוחים, ומחליטים האם נדרש שינוי באיפיון המערכת, עקב גילוי באגים** | **Bug meeting** |
| **QA, DEV** | **כאשר אין עילה לבאג, בד"כ נובע מחוסר הבנה** | **Reject** |
| **QA** | **כפל באגים, בד"כ ייסגרו לפני הישיבה** | **Duplicated** |
| **QA only** | **פתיחת באג חדש במערכת** | **Open bug** |
| **QA** | **לאחר שהבאג עבר תיקון אצל ה- DEV, מוחזר ל- QA, ומתבצעות בדיקות אימות ורגרסיה** | **Bug resolved** |
| **QA, DEV** | **ה- QA מצא כי הבאג לא נפתר, ומחזיר את הבאג ל- DEV.** | **Reopen test meeting** |
| **QA** | **אימות ע"י ה- QA, כי הבאג אכן תוקן, ולא נוצרו באגים חדשים בעקבות התיקון** | **Bug verification** |
| **QA only** | **הבאג נסגר** | **Bug is closed** |
| **QA, DEV** | **הבאג מועבר ל- DEV בצורה שאינו מצליח לשחזר אותו, לכן הוא מחזיר אותו ל- QA בסטאטוס Reproduce.**  **על ה- QA לשחזר את הבאג ולפתוח אותו מחדש.**  **במידה ולא מצליחים לשחזר את הבאג, הבאג נסגר.** | **Reproduce** |
| **QA, DEV, PM** | **הטיפול בבאג נדחה לגרסה הבאה ,מטעמי חוסר דחיפות** | **Postponed to next release** |
| **DEV** | **תיקון הבאג דורש תחקור מעמיק – באג מורכב.**  **יש לבצע מעקב לשם שחזור, יכול להמשך זמן ארוך מהרגיל – כשבוע - שבועיים** | **Monitor** |

**5. קריטריונים לאישור המערכת**

|  |  |  |
| --- | --- | --- |
| **בפועל** | **יעד** | **קריטריון** |
| **100%** | **100%** | **% בדיקות Sanity שעברו בהצלחה** |
| **78%** | **70%** | **% בדיקות שבוצעו מתוך בדיקות שתוכננו** |
| **95%** | **85%** | **% בדיקות שעברו בהצלחה מתוך בדיקות שבוצעו** |
| **0%** | **5%** | **% בדיקות בסטטוס “Blocked”** |

**טסטים:**

**באגים:**

|  |  |  |
| --- | --- | --- |
| **בפועל** | **יעד** | **קריטריון** |
| **0** | **0** | **באגים פתוחים ברמת חומרה Critical** |
| **0** | **0** | **באגים פתוחים ברמת חומרה High** |
| **0** | **10%** | **באגים פתוחים ברמת חומרה Medium** |
| **2%** | **20%** | **באגים פתוחים ברמת חומרה Low** |

**6. מסקנות והמלצות**

**לסיכום, כמתואר במסמך זה, המערכת עברה את מירב הבדיקות בהצלחה, ועמדה ביעדים שהוקצו.**

**אנו ממליצים לשחרר את המערכת לשלב הייצור ומשם ללקוח.**

**חלק מהבאגים הקיימים, שטרם תוקנו, אינם מפריעים לעבודת המערכת התקינה,**

**יש לתקנם בעתיד, עיקר הבאגים שנותרו הם בממשק המשתמש.**

**בשלב הבא, נמליץ לפתח מערכת מותאמת למערכות הפעלה של IOS ,Apple כרגע עבור לקוחות קבועים התוכנה עובדת רק באנדרואיד.**

**בהמשך אפילו כדאי לחשוב על כדאיות לפתח מערכת זו גם עבור כלל הלקוחות , ולא רק ללקוחות קבועים.**