

סוגי בדיקות

כל סוג בדיקות בודק היבט מסויים במערכת.
הדוגמאות מבוססות על מסמך האפיון של דיווח שעות שמוריה שלחה בקבוצה.

1. בדיקות ממשק משתמש (UI)

כל מסמך שאנחנו רואים והוא משקף לנו מידע זה UI.
מתייחס להוראות שהמשתמש נותן והתגובות של המערכת UI = User Interface
מתייחס לתוכן של המסך GUI = Graphical User Interface

אין בבדיקות BATCH בדיקות ממשק משתמש כי שם אין UI.
בדיקת התוכן על המסך, הוראות, ללא התייחסות למהות המסך.
בדיקה צורה כללית, צבעים, פונטים, טקסט וכדומה.
בדיקת שדה - אני יכול להקליד בשדה נתונים בפורמט כלשהו, למשל תווים (מספרים, אותיות, סימנים). למשל, בשדה שנועד רק למספרים אנסה להקליד סימנים או אותיות.

דוגמא:

כותרת (מערכת רישום עובד חדש)

[הוראות לכל שדה ממסמך האפיון]	[השדות השונים]
שם עד 20 תווים, NOT NULL (אי אפשר להשאיר ריק)	<input type="text"/>
ת.ז. INT (אפשר להכניס מספרים שלמים בלבד) PK (אני לא יכול להכניס את אותו נתון שכבר קיים)	<input type="text"/>
תאריך לידה DATE (רק בפורמט תאריך מסויים)	<input type="text"/>
משכורת MONEY (חייב להיות גדול מ-0)	<input type="text"/>

כאשר אני עושה בדיקת UI לא מעניין אותי מה המהות של המערכת ומה היא עושה. אלא שהשדות או המערכת עצמה מבצעת את מה שמוטל עליה לבצע. כלומר שהכפתור הספציפי מבצע את מה שהוא אמור לבצע, שהשדות מתריעות במידה והוכנס ערך לא תקין. למשל, בבדיקה זו אני אנסה לשחק עם השדות ולבחון את המגבלות שלהן.

השדות השונים שקיימים במערכות תוכנה:

- א. שדה נומרי - לפעמים יהיה כתוב INT לפעמים אפילו רק N במסמך האיפיון כדי להבין שזה שדה נומרי. נרצה לדעת האם הוא יכול לקבל מספר עשרוני או שלם. נרצה להקפיד לבדוק גבולות (מינוס, אפס, מספר עם כמות תווים מקסימליים, מספר שלילי).
- [אגב, באופן כללי הספרה 0 היא ספרה בעייתית, היא עושה בעיות פעמים רבות, לכן ספרה זו תהווה עבורנו פעמים רבות גבול]. נבדוק את הגבולות גם מבחינת מספר תווים.
- ב. שדות תווים ומספרים (Alpha Numeric) - שדות שמקבלים גם אותיות וגם מספרים. נבחן את מגבלת כמות התווים, האם ניתן להשאיר רק או לא, אותיות בשפות אחרות.
- ג. שדה תאריך - נבחן את התאריך בפורמטים שונים. מה קורה בגבולות. מה אם אני שם 40/08/2021 למשל? מה אם אני שם שנה עתידית? אבדוק את כל הגבולות.

דוגמא ממערכת דיווח שעות:

הקלדת ערכים בשדות "משעה" ו- "עד שדה"

השדה והפורמט	מילוי השדה בפועל	תוצאה
שעה (HH) באותיות	AB:00	שלילי
שעה = 27	27:00	שלילי
שעה = NULL	____:30	שלילי
דקות (MM) = באותיות	08:CD	שלילי
דקות = 78	08:78	שלילי
זמן תקין	08:30	חיובי
דקות = NULL	08:____	שלילי

כמו כן, ננסה גם להציב מספרים שליליים.

בדיקות חיוביות: אני מכניס ערכים תקינים והמערכת מגיבה בדרך שנועדה להגיב בה, היא מקבלת את הערכים כתקינים.

בדיקות שליליות: כאשר אני מכניס ערכים שגויים ומנסה לראות שהמערכת מגיבה כפי שנועדה להגיב, כלומר, מתריעה ולא נותנת להמשיך. אצפה אפילו שהיא תאמר לי מה היא הטעות שלי.

המשך סוגי השדות השונים:

ד. כפתור בחירה (Radio Button) - לוחצים על הכפתור ולרוב זה מאפשר רק בחירה אחת מתוך האופציות. נרצה לוודא שזה אכן נותן לבחור אופציה אחת. כמו כן, נרצה לוודא שזה נותן ברירת מחדל שכבר מסומנת, בדרך כלל האופציה הנפוצה ביותר, כדי להקל על המשתמש.
ה. תיבת סימון (Check Box) - מאפשר בדרך כלל בחירה מרובה. יכולה להיות לפעמים לוגיקה - למשל אם אני אוריד V מתיבה מסויימת אז זה יכבה לי אפשרות לסמן V בתיבות אחרות ולהפך. אבדוק שאני יכול לבחור הכל, אם אחד תלוי באחר אז לבדוק שהתלות קיימת.
ו. תיבת רשימה נפתחת (Combo Box) - לוחץ על כפתור ונפתח לי תפריט ברשימה שבו אני צריך לבחור אופציה. בדרך כלל כאשר יש הרבה אופציות. נעדיף שזה יהיה מסודר בדרך מסויימת. לפעמים צריכה להיות אופציה של "סמן הכל" והוא בוחר הכל. אז אבדוק שכל הדברים האלה קיימים ועובדים.

2. בדיקות פונקציונליות

כאן רוצים לבדוק את כל הפעולות שהמערכת מבצעת. אם למשל אני מכניס שם משתמש וסיסמא (קלט) אז הפלט שלי זה שעברתי למסך הבא.
אנחנו נחלק את המערכת לרשימת הפעולות שלה (יש לכך דרכים) ואז נבדוק כל פעולה בודדת בנפרד.
על כל פעולה כזו נבצע בדיקות חיוביות וגם בדיקות שליליות.

דוגמא - פעולות ב"דיווח שעות":

הכנסת שעות - האם יש סיכום של השעות

האם המנהל יכול לצפות בשעות (פירוט וסיכום)

האם אפשר לערוך את השעות

האם כפתור שמירה פעיל

האם הכניסה למערכת עובדת

האם אפשר לדווח על שעות

אנחנו ניקח כל פעולה ופעולה כזו ונבדוק אם אנחנו מקבלים את התוצאה המיועדת.

אז במערכת דיווח שעות נוודא גם את הפעולות הבאות:
אפשר לעשות דיווח שעות תקין - משעה 08:00 עד שעה 17:00 למשל
דיווח בשבת וימי חג
להכניס אותן שעות
להפוך בין השעות
לדווח רק על שדה אחד, בלי השני
דיווח מספרים לא עגולים (08:37 עד 17:24)
דיווח מעל 20 שעות

3. בדיקות תהליכים

נרצה לבדוק אוסף של פעולות ברצף. יכול להיות שפעולות עובדות בנפרד, אך ביחד וברצף משהו יתגלה כלא תקין.
נקרא לפעמים לבדיקה זו END-TO-END או E2E.

לדוגמא: דיווח שעות למספר ימים אחורה, עדכון דיווח שעות קודם, דיווח שעות ע"י עובדים ובמקביל צפיית המנהל בדיווח שעות העובדים.
או - מעדכן שעה, שמירה, מעבר לחודש אחר, עדכון דיווח קיים, שמירה (כלומר, יש פה תהליך של פעולות שונות).

זה משמש גם עבור בדיקות רגרסיה בסיום הבדיקות המתוכננות.
הרי לפני מסירת המערכת ללקוח עושים עוד בדיקות אחרונות, של רגרסיה ולכן נשתמש גם בבדיקות תהליכים.
חושבים על פעולות שמשתמש באמת יבצע. אפשר גם לחשוב על פיולות שמתנגשות זו בזו, למשל כאשר 2 עובדים מדווחים במקביל והמנהל צופה בזה בזמן, או אפילו ממש פעולות שדורסות זו את זו.

4. התאוששות מנפילות

זה היבט של המערכת שפעמים רבות לא מתייחסים אליו כשכותבים את האפיון. בודקים איך המערכת מגיבה כאשר המערכת נופלת, למשל נפילת שרת, הפסקת חשמל.

הולכים לבנק, פותחים חשבון, הבנקאי עובר ממסך למסך, תהליך ארוך. פתאום הפסקת חשמל, מערכות נופלות. חוזר החשמל, מעלים את המערכת - מה תיהיה תגובת המערכת? האם הנתונים נשמרים?

לכן, בבדיקה זו נפיל את המערכת בצורה מלאכותית, נעלה את הכל מחדש ונראה מה המערכת שמרה או לא שמרה.

לדוגמא: נדווח שעות למספר ימים אחורה, לא נלחץ על כפתור "שמירה", נפיל את המערכת באופן מלאכותי, נעלה בחזרה את המערכת. נצפה שמשהו יישמר. שתיהיה כל רגע שמירה של הנתונים - נגיד כל 5 דקות, או כל 5 שורות או במעבר ממסך למסך וכיו.

5. בדיקות גיבוי ושיחזור

בהרבה חברות, בעיקר כאלה שמבוססות על מאגרי נתונים, המערכת מגבה את הנתונים כל כמה זמן, למשל כמה פעמים ביום, פעם ביום (כמו למשל בבנקים, בלילה). למשל, אם נהרסו הנתונים בשל באגים חמורים או תקיפות סייבר אז אנחנו רוצים את האופציה שיהיה ניתן לשחזר את הנתונים. יש מערכת שמגבה ומשחזרת וגם מערכת זו צריכה להיבדק. כלומר, יש לנו את מאגר המקור של הנתונים (למשל, אם זה בנק אז החשבונות הקיימים) ויש מאגר גיבוי. לאחר מכן יש פעולת שחזור ממאגר הגיבוי ומקבלים מאגר משוחזר. אנחנו רוצים לוודא שמאגר המקור דומה למאגר המשוחזר.

6. בדיקות התקנה

בדיקות עבור תוכנות שאנחנו מתקינים. בודקים שתהליך ההתקנה הוא ברור ופשוט (במיוחד כשמדברים על אפליקציות). לאחר תהליך ההתקנה נפתח את המערכת, נבצע כמה פעולות בסיסיות וזאת כדי לבדוק שהמערכת עובדת לאחר ההתקנה בצורה תקינה. נרצה לבדוק התקנה על מחשב STAND ALONE, על שרת, אם זה משהו אינטרנטי אז בדפדפנים שונים, על מערכות הפעלה שונות.