**Tesztelési terv**

|  |  |
| --- | --- |
| **Dokumentum címe: (azonosítója)** | Tesztelési terv |
| **Minősítés: (állapot)**  *(tervezet, jóváhagyott, stb.)* |  |
| **Verziószám:** | v1.0 |
| **Projekt név:** | Vállalatirányítási rendszer |
| **Készítette:** | Szabolcs Gábor |
| **Telefon:** |  |
| **Utolsó mentés kelte:** | 2018. 01. 12. |
| **Dokumentum célja:** |  |
| **Fájlnév:** | Tesztelési terv.docx |

**A dokumentum célja:**

A tesztelési terv célja a tesztelés teljes körűségének a biztosítása, a tesztelés során alkalmazott eljárásik és megoldások meghatározása.

**Tesztelés menete:**

A megbeszéltek alapján minden fejlesztő a rá kiosztott folyamatot teszteli. A teszteket több alkalommal is el kell végezni, hogy csökkentsük annak az esélyét, hogy csak egy véletlen folytán kaptunk jó eredményt a folyamat alatt. A tesztek eredményét minden alkalommal jegyzőkönyvezni kell!

**Határidők**:

A tesztelések határidejét úgy állapítsuk meg, hogy a fejlesztőnek legyen még elegendő ideje korrigálni az esetlegesen felmerülő hibákat.

**Tesztelés** **folyamat**:

Határozzuk meg hogy az adott tesztel milyen célt akarunk elérni.  
 A cél minden egyes kimeneteli lehetőségét vizsgáljuk ki, Általános funkcionális teszt és Szélsőérték teszt módszerének az alkalmazásával.  
 Az elért eredményeket foglaljuk Tesztelési jegyzőkönyvbe

**Tesztelési típusok**:

**Funkcióteszt:**

**Általános funkcionális teszt:** A tesztelés során pontos adatokkal dolgozunk, azt feltételezve, hogy az általunk bevitt adatok pontosak, és ez alapján az elvárt eredmény a tökéletes működés legyen.

**Szélsőérték teszt:** A tesztelés során fals adatokkal dolgozzunk, ezáltal vizsgálhatjuk az esetleges felhasználói tévedéseket, elírásokat. A tesztelés kimeneteleként az elvárt eredmény, hogy valamilyen akadályozó tényezőbe ütközzünk, pl.: sikertelen bejelentkezés, sikertelen megrendelés, stb. Ha a fals eredmény mégis hiba nélküli eredményt hoz, akkor azt felül kell vizsgálni, hogy a tesztelés során volt-e a hiba, vagy pedig a programban.

**Biztonsági teszt**: A biztonsági tesztek azt vizsgálják, hogy a rendszer mennyire sebezhető rosszindulatú támadás vagy használat esetén. E témakörbe tartozik adatok védelme és a hálózati környezet biztonságossága, illetve a rendszer adatbázisaiban tárolt érzékeny adatok titkosítása, mint például a jelszavak.:  
 **Sebezhetőség vizsgálata:**  
 Automatikus tesztezközökkel keresünk biztonsági hibákat.  
 **Behatolási teszt:**  
 A behatolási teszttel egy rosszindulatú felhasználó tevékenységét szimuláljuk, lehet black-box és white-boksz megközelítéssel is vizsgálni.

**Teljesítményteszt:**

**Terheléses teszt:** A terheléses tesztek célja, hogy bizonyos terhelés mellett keressük a rendszer szűk keresztmetszeteit.  
 Lehetséges szűk keresztmetszet:  
 - rendszerbe való be- és kijelentkezés,  
 - az asztali és webes program közötti szinkronizáció (megrendelések).

**Stresszteszt:** Célja, hogy megmutassa: a rendszer stabilitása az elvárásoknak megfelelő.  
 Az elvárttól nagyobb terhelés alatt működtetjük a rendszert és vizsgáljuk az egyes elemeket.

**Mennyiségi teszt:** A rendszert nagy adatbázison, nagy rekordszám mellett vizsgáljuk, figyeljük, hogy hiányos, vagy nem megfelelő indexelés mellett is stabilan működik a rendszer az adatbázis méretétől függetlenül.

**Főbb tesztelendő folyamatok:**

**Asztali alkalmazás:** - bejelentkezés (Általános funkcionális teszt, Szélsőérték teszt),  
 - keresés,  
 - termék felvitele (Általános funkcionális teszt, Szélsőérték teszt),  
 - termék módosítás (Általános funkcionális teszt, Szélsőérték teszt),  
 - számla kiállítása,  
 - megrendelések szinkronizálása,  
 - kijelentkezés (Terheléses teszt).

**Webes felület** - bejelentkezés (Általános funkcionális teszt, Szélsőérték teszt),  
 - szűrési funkció (Általános funkcionális teszt, Szélsőérték teszt),  
 - regisztráció (Általános funkcionális teszt, Szélsőérték teszt),  
 - kosár,  
 - adatmódosítás (Általános funkcionális teszt, Szélsőérték teszt),  
 - megrendelés,  
 - kijelentkezés.