**Software testing.**

**Smart home.**

*1. Application description*

Inteligentní simulace domu, kdy simulujeme provoz domácnosti, využíváme jednotlivá zařízení domu a vyhodnocujeme využití, spotřebu, volnou a pracovní dobu jednotlivých osob

Spustí se hlavní třída, která vytvoří simulaci s konfigurací zvolenou uživatelem. Uplyne celý rok a zprávy jsou generovány v jednom textovém souboru.

*2. Testing strategy*

|  |  |  |  |
| --- | --- | --- | --- |
| **Quality**  **characteristic** | **Proces** | **Požadavek** | **Část systému** |
| Bezchybá funkcionalita | Krmení domací zvíře | Volný člověk ve stejném pokoji krmí zviře ,když má hlad | Pet simulation |
|  | Změna pokoje | Nahodně mění pokoj | People simulation |
|  | Generování akce | Každé pul hodiny se generují akce | Action simulation |
|  | Report write | Report má spravný čas | Report generation |
|  | Oprava závad | Závady smi opravovat pouze otec a dědeček | People simulation |
|  | Stárnutí zařezení | Snižuje se kvalita | House simulation |
|  | Doručení jídla | Courier každý den doručí jídlo | House simulation |
|  | Změna počasí | Počasí souvisí se současnou dobou | House simulation |
|  | Aktualizace spotřeby | Každý den se zvyšuje hodnota aktuální spotřeby podle typu zařizení | House simulation |
|  | Meření | Čidla meří hodnoty(vlhkost, svět atd.) podle typu zařizení | House simulation |
|  | Změna stavu | Zařizení se mění stav(ok/broken) dle aktualní kvality | House simulation |
|  | Změna stavu | Obyvateli se mění stav(sleeping/awake)  dle času | House simulation |
|  | Krmení ditě | Maminka krmí ditě když ditě má hlad | People simulation |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Proces** | **Požadavek** | **Možné poskození** | **Vysvětlení poskození** | **Část systemu** | **Pravděpodobnost selhaní** | **Vysvětlení**  **pravděpodobnosti** | **Třida rizika** |
| Krmení domací zvíře | Volný člověk ve stejném pokoji krmí zviře ,když má hlad | L | Zvíře bude mít negativní satiety | Pet simulation | L | V pokoji není žadný volný člověk | C |
| Změna pokoje | Lide nahodně mění pokoji | H | Zacyklení simulace | People simulation | L | Všichni ubytovateli nebudou měnit pokoj | B |
| Generování akce | Každé pul hodiny se generují akce | H | Simulace nemá smysl | Action simulation | L | List volných použitelných zařizení je prazdný | B |
| Report write | Report má spravné udaje | H | Nesmyslné reporty | Report generation | L | Jedna akce popisovaná několika reporty | B |
| Oprava závad | Závady smi opravovat pouze otec a dědeček | M | Zařizení zustane rozbité | People simulation | L | Otec a dědeček budou zaneprazdněné | C |
| Stárnutí zařizení | Během času se snižuje kvalita | M | Zařizení nebude měnit stav | House simulation | L | Hodnota odebrané kvality nepatři dánemu typu zařizení | C |
| Doručení jídla | Courier každý den doručí jídlo | L | Lednička se nebude plnit | House simulation | L | Fridge is null | C |
| Změna počasí | Počasí souvisí se současnou dobou | H | Čídla budou měřit špatné udaje | House simulation | L | Chyba určení doby podle aktuálního času | B |
| Aktualizace spotřeby | Každý den se zvyšuje hodnota aktuální spotřeby podle typu zařizení | M | Consumption reporty nebodu davat smysl | House simulation | L | Hodnota přidané spotřeby nepatři dánemu typu zařizení | C |
| Meření | Čidla meří hodnoty(vlhkost, svět atd.) podle typu zařizení | M | Reporty nebudou davat smysl | House simulation | L | Simulace se nemění počasí | C |
| Změna stavu | Obyvateli se mění stav(sleeping/awake)  dle času | L | Nebudou se generovat reporty spaní a probouzení | House simulation | L | Obyvatel zasekne v nějakém stavu | C |
| Změna stavu | Zařizení se mění stav(ok/broken) dle aktuální kvality | M | Nedavalo by smysl | House simulation | M | Kvalita se nebude měnit během času | B |
| Krmení ditě | Maminka krmí ditě když ditě má hlad | L | Ditě bude mit negativní nasycení | People simulation | L | Maminka is null | C |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Funkce** | **Třida rizika** | **Revize** | **Vývojářské testy** | **Systémové testy** | **UAT** | **Test v produkci** |
| **Bezchybná funkcionatlita** | | | | | | |
| Změna pokoje | B |  |  |  |  |  |
| Generování akce | B |  |  |  |  |  |
| Report write | B |  |  |  |  |  |
| Změna počasí | B |  |  |  |  |  |
| Změna stavu | B |  |  |  |  |  |
| Krmení domací zvíře | C |  |  |  |  |  |
| Oprava závad | C |  |  |  |  |  |
| Stárnutí zařizení | C |  |  |  |  |  |
| Doručení jídla | C |  |  |  |  |  |
| Aktualizace spotřeby | C |  |  |  |  |  |
| Meření | C |  |  |  |  |  |
| Změna stavu | C |  |  |  |  |  |

**Třidy ekvivalence vstupů**

Pro analýzu třid ekvivalence volíme config JSON file ve kterém je konfigurace obyvatelů(Mom, Dad, Children,

Grandpa, Grandma, Babies, Cats, Dogs,Parrot). Mom, Dad, Grandpa, Grandma, Parrot mají 2 parametry: Age a Name. Children, Babies, Cats, Dogs jako parametr mají list parametrů taky obsahující Age a Name.

Niže je tabulka s analyzou těchto parametrů pro všechny obyvately.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parametr** | **Název třidy ekvivalence** | **Validita třidy ekvivalence** | **Hodnoty třídy ekvivalence** | **Mezí podmínky** |
| Name | validName | Validní EC | String | String začinající velkým písmem |
| invalidName | Nevalidní EC | Stringy |
| emptyName | Nevalidní EC | String |
| typeErrorName | Nevalidní EC | Jiný než typ String |
| Age | parentsAge | Validní EC | <30;50> |  |
| invalidParentsAge | Nevalidní EC | <-2,147,483,648;30)  {\displaystyle \wedge } ∧ (50; 2,147,483,647 > |  |
| childrenAge | Validní EC | <4; 18> |  |
| invalidChildrenAge | Nevalidní EC | <-2,147,483,648;4)  {\displaystyle \wedge }∧ (18; 2,147,483,647 > |  |
| babiesAges | Validní EC | <0;4) |  |
| invalidBabiesAge | Nevalidní EC | <-2,147,483,648;0)  {\displaystyle \wedge }∧ <4; 2,147,483,647 > |  |
| grandparentsAge | Validní EC | (50;100> |  |
| invalidGrandparentsAge | Nevalidní EC | <-2,147,483,648;50>  {\displaystyle \wedge }∧ (100; 2,147,483,647 > |  |
| catsAge | Validní EC | <0;15> |  |
| invalidCatsAge | Nevalidní EC | <-2,147,483,648;0)  {\displaystyle \wedge }∧ (15; 2,147,483,647 > |  |
| dogsAge | Validní EC | <0;20> |  |
| invalidDogsge | Nevalidní EC | <-2,147,483,648;0)  {\displaystyle \wedge }∧ (20; 2,147,483,647 > |  |
| parrotAge | Validní EC | <0;10> |  |
| invalidParrotAge | Nevalidní EC | <-2,147,483,648;0)  {\displaystyle \wedge }∧ (10; 2,147,483,647 > |  |
| typeErrorAge | Nevalidní EC | Jiný než typ integer |  |

Pro další analýzu třid ekvivalence volíme metodu StuffAPI.useStuff(person, time , stuff)

Tato metoda vytvaří akce Use pro určité použitelné zařizení.

Její tři parametry jsou:

* Person je člověk který použivá zařizení
* Time je čas (minuty) Use akce
* Stuff je použivané zařizení

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parametr** | **Název třidy ekvivalence** | **Validita třidy ekvivalence** | **Hodnoty třídy ekvivalence** | **Mezní podmínky** |
| Person | Mom | Validní EC | Mom |  |
| Dad | Validní EC | Dad |
| Grandma | Validní EC | Grandma |
| Grandpa | Validní EC | Grandpa |
| Child | Valdidní EC | Child |
| Baby | Nevalidní EC | Baby |
| Courier | Nevalidní EC | Courier |
| RepairServiceMaster | Nevalidní EC | RepirServiceMaster |
| nullPerson | Nevalidní EC | nullType |
| typeErrorPerson | Nevalidní EC | typeErrorType |
| Time | validTime | Validní EC | <0;525600> |  |
| invalidTimePositive | Nevalidní EC | (525600;2,147,483,647) |
| invalidTimeNegative | Nevalidní EC | <**-**2,147,483,648;0) |
| Stuff | Bike | Validní EC | Bike |  |
| Car | Validní EC | Car |
| Computer | Validní EC | Computer |
| Fridge | Validní EC | Fridge |
| Ski | Validní EC | Ski |
| TV | Validní EC | TV |
| nullStuff | Nevalidní EC | null |
| typeErrorStuff | Nevalidní EC | Jiný typ než Stuff |

1)Měření Lamp

A = updateSensors

B = measureSensors

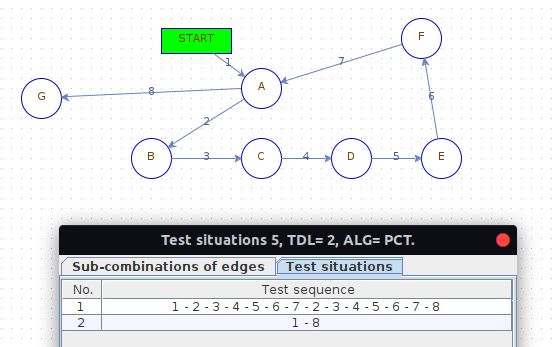
C = meauserLamp

D = getCurrentLightLevel

E = ElectricityConsumptionDay + 0.2

F = changeState

G = don’t change

 1 = 2 = 3 = 4 = 7 = nic

5 = isTurnOn

6 = currentLightLevel < 30 and hours >= 8 and not is Lightning

8 = night is now

2)Change Room

A = Choose Inhabitants

B = Stop Use Staff

C = Person Change Room to Bedroom

D = Stop Use Staff

E = Baby Change Room to Bedroom or LivingRoom

F = Person Change Room

1 = nic

2 = hourTime >= 0 and hourTime <= 8

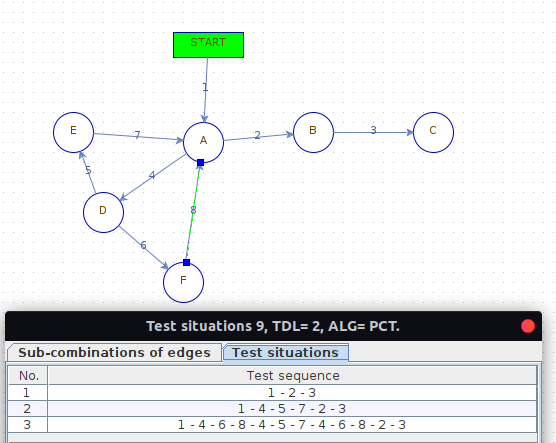
3 = inhabitant is a Person

4 = hourTime >= 8 and hourTime <= 23

5 = inhabitant is a Baby

6 = inhabitant is not a Baby

7 = 8 = hourtime + 30



|  |  |
| --- | --- |
| **Parameter** | **Obsah** |
| Nazev testu | DadRepairStuffTrue |
| Hloubka test | Střední |
| Vstupni podminki | Spuštění simulace |
| Popis testu | 1)Získáme tatínka v domě  2)Najdeme televizi  3)Nastavíme hodnotu qualtity televize na min a stuffState na Broken  4)Tatínek získá dokumentaci přečte ji a opraví televizi  5)tvMock.checkQuality vrátí true pravě tehdy když televize bude opravena |
| Ocekavany vysledek | Funkce tvMock.checkQuality() vrátí true |
| Testovaci data | Dad, TV |

|  |  |
| --- | --- |
| **Parameter** | **Obsah** |
| Nazev testu | AllPeopleSleepInBedroom |
| Hloubka test | Střední |
| Vstupni podminki | Spuštění simulace |
| Popis testu | 1)Spustíme simulaci  2)Dostaneme všechny reporty za 3 dny  3)Odfiltrujeme všechny reporty a dostaneme reporty bez Food Courier a se všemi personami  4)Zkontrolujeme aby mezi 0 a 7 hodinami všechny lidi byli v Bedroom |
| Ocekavany vysledek | False, protože tatínek muže něco opravovat v tento čas |
| Testovaci data | Person, Report, Simulation |