

1. Návrh testovací strategie

1.1. Popis aplikace

1.1.1. Popište funkcionalitu aplikace.

Testovaná aplikace je implementována jako semestrální práce v rámci předmětu B6B36OMO, úkolem které je vytvoření virtuální simulace inteligentního domu, což zahrnuje chod domácnosti s ohledem na volný a pracovní čas jednotlivých osob, použití různých zařízení a vyhodnocení jejich využití a spotřeby.

1.2. Testovací strategie

1.2.1. Vytvořte přehled částí aplikace.

Část systému	Proces	Požadavek	Test goal
Správa domu	Založení domu	Načítají se předem definované pokoje, lidé a domácí mazlíčky, zařízení a další přístroje.	<i>chybí</i>
Krok simulace	Generace událostí v pokoji	Pro každý pokoj v domě se s pravděpodobností 10 % náhodně generuje událost.	V pokojích se náhodně generují události a pak jsou odbavovány vhodnou osobou.
	Aktualizace parametrů pokoje	Po každé iteraci se aktualizují pokoje a zařízení, které do nich patří.	<i>chybí</i>
	Aktualizace osob a domácích mazlíčků	V každé iteraci osoby a domácí mazlíčky vykonávají náhodně generované akce. Podle vybrané akce probíhá následná aktualizace.	Jednotlivé osoby a domácí mazlíčky mohou provádět akce, které mají efekt na zařízení.
	Generace informací o domě	Na konci iterace se ukládá aktuální stav domu zahrnující teplotu v pokojích, které mají teploměr, a aktuální spotřebu elektřiny v celém domě s její přibližnou cenou.	V každé iteraci se aktualizuje spotřeba elektřiny a funkčnost jednotlivých zařízení a může se měnit teplota v pokojích.
Přehled akcí	Generace reportu	Vytvářejí se tři .txt soubory obsahující aktivitu jednotlivých osob a zvířat, seznam událostí podle typu, zdroje a cíle, množství spotřebované elektřiny jednotlivými zařízeními.	Jsou generovány reporty událostí, aktivit a spotřeby elektřiny.

1.2.2. Prioritizujte části aplikace.

Proces	Možné poškození	Vysvětlení možného poškození	Pst. selhání	Vysvětlení pravděpodobnosti selhání	Třída rizika
Založení domu	H	V případě, že dům nebude správně generován, bude celá aplikace nepoužitelná.	L	Je to jednoduchá část kódu, která používá předem definované objekty. Problém se může objevit pouze v případě chyby ze strany programátora.	B
Generace událostí v pokoji	M	Může nastat situace, že obyvatelé nebudou informováni o tom, že došlo k nějaké akci, která by mohla způsobit vážné poškození celému domu, anebo budou informováni o neexistující události.	M	Vysoká frekvence používání (v každé iteraci se generují náhodné události ve všech pokojích). Závislost na stavu jednotlivých pokojů.	B
Aktualizace parametrů pokoje	M	Špatně aktualizované parametry pokoje mohou způsobit generování přebytečných událostí, jejich nedostatek anebo kolize.	L	Díky jednoduché implementaci a nezávislosti na ostatních objektech programu je tato část kódu úplně bezpečná, i když jsou parametry pokojů aktualizovány v každé iteraci.	C

Aktualizace osob a domácích mazlíčků	M	Špatně aktualizované parametry osoby nebo domácího mazlíčka mohou způsobit výběr více než jedné akce v jedné iteraci. Tím vznikne situace, která poruší logiku zadání.	H	Vysoká frekvence používání (v každé iteraci se aktualizují parametry všech osob a domácích mazlíčků). Nejsložitější z programátorského pohledu část kódu. Vysoká závislost na stavech ostatních objektů.	B
Generace informací o domě	M	Informace o jednotlivých pokojích v domě může být chybně uvedena.	M	Závislost na jenom jednom objektu, který nemá velice náročné metody. Velká frekvence používání (v každé iteraci).	B
Generace reportu	L	Díky tomu, že reporty se generují jen na konci simulace, můžeme být zcela jistí, že nebudou mít žádný vliv na nejdůležitější části programu. Ale mohou se generovat reporty, které obsahují neexistující data.	M	Kvůli přímé závislosti na náročných ve smyslu implementace objektech, data poskytnuta od nich pro uložení v reportech může obsahovat chyby.	C

1.2.3. Vypracujte test levels.

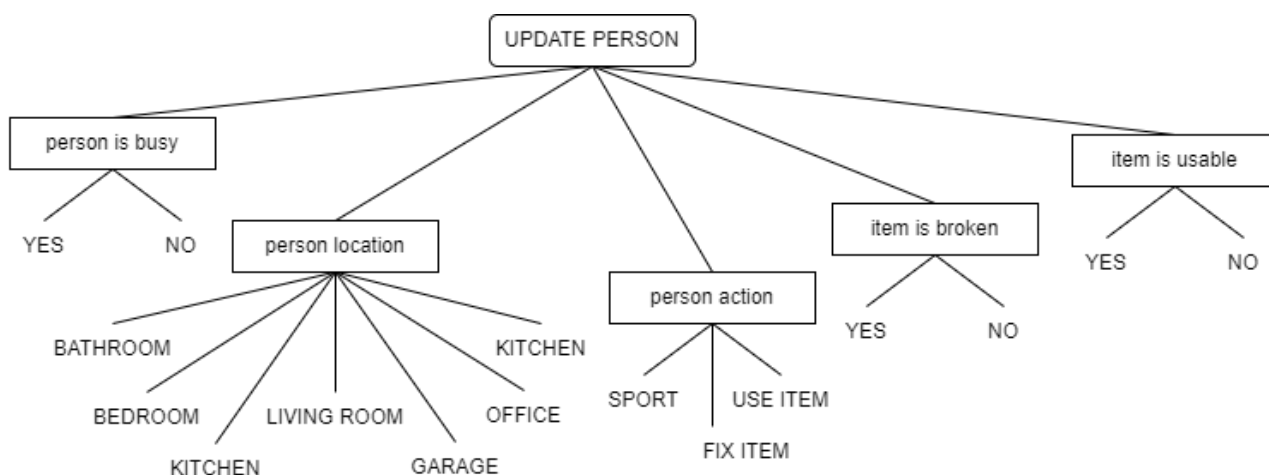
Část systému / funkce	Třída rizika	Test levels				
		Revize	Vývojářské testy	Systémové testy	UAT	Test v produkci
Správa domu						
Založení domu	B	ano	střední	nízká	střední	
Krok simulace						
Generace událostí v pokoji	B		nízká	střední		
Aktualizace parametrů pokoje	C			střední	nízká	
Aktualizace osob a domácích mazlíčků	B			vysoká	střední	
Generace informací o domě	B		střední	střední	střední	ano
Přehled akcí						
Generace reportu	C		střední	nízká	vysoká	ano

2. Testovací scénáře

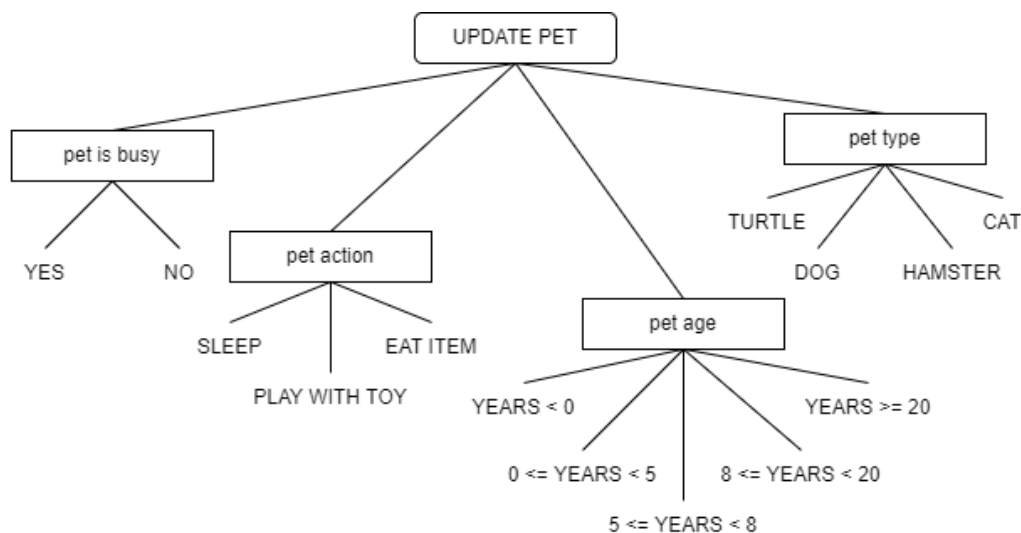
2.1. Testy vstupů

2.1.1. Pro každý vstupní parametr analyzujte EC a určete mezní podmínky (pokud možno).

Prvním vstupem aplikace je signatura metody zodpovědné za výběr akce, kterou bude vykonávat osoba v aktuální iteraci. Parametry jsou:



Druhým vstupem aplikace je signatura metody zodpovědné za výběr akce, kterou bude vykonávat domácí mazlíček v aktuální iteraci. Parametry jsou:



Mezní podmínky je možné určit pouze pro věk domácího mazlíčku příslušný intervalové třídě ekvivalence, to jsou čísla -1, 0, 1, 4, 5, 6, 7, 8, 9, 19, 20, 21.

2.1.2. Vytvořte kombinace testovacích dat technikou pairwise testing.

Kombinace testovacích dat odpovídající prvnímu vstupu:

Person is busy	Person location	Person action	Item is usable	Item is broken
No	Bathroom	Use item at home	No	Yes
No	Kitchen	Use item at home	Yes	No
Yes	Bedroom	Use item at home	Yes	Yes
Yes	Bathroom	Fix item	No	No
Yes	Office	Sport	Yes	Yes
Yes	Garage	Use item at home	No	No
Yes	Kitchen	Fix item	Yes	Yes
Yes	Living room	Sport	No	No
No	Bathroom	Sport	No	Yes
No	Office	Use item at home	Yes	No
No	Garage	Fix item	No	Yes
No	Kitchen	Sport	Yes	No
No	Living room	Use item at home	No	Yes
No	Bedroom	Fix item	Yes	No
Yes	Office	Fix item	Yes	Yes
Yes	Garage	Sport	No	No
Yes	Kitchen	Use item at home	Yes	Yes
Yes	Living room	Fix item	No	No
Yes	Bedroom	Sport	Yes	Yes
Yes	Bathroom	Use item at home	No	No
No	Garage	Use item at home	No	Yes
No	Kitchen	Fix item	Yes	No
No	Living room	Sport	No	Yes
No	Bedroom	Use item at home	Yes	No
No	Bathroom	Fix item	No	Yes
No	Office	Sport	Yes	No

Kombinace testovacích dat odpovídající druhému vstupu:

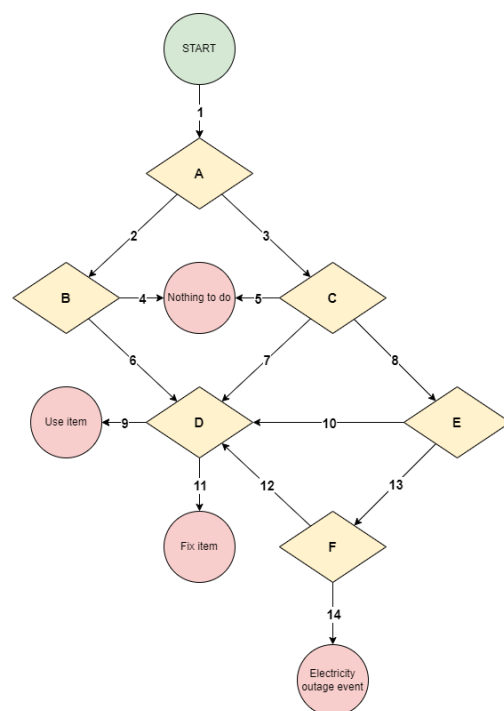
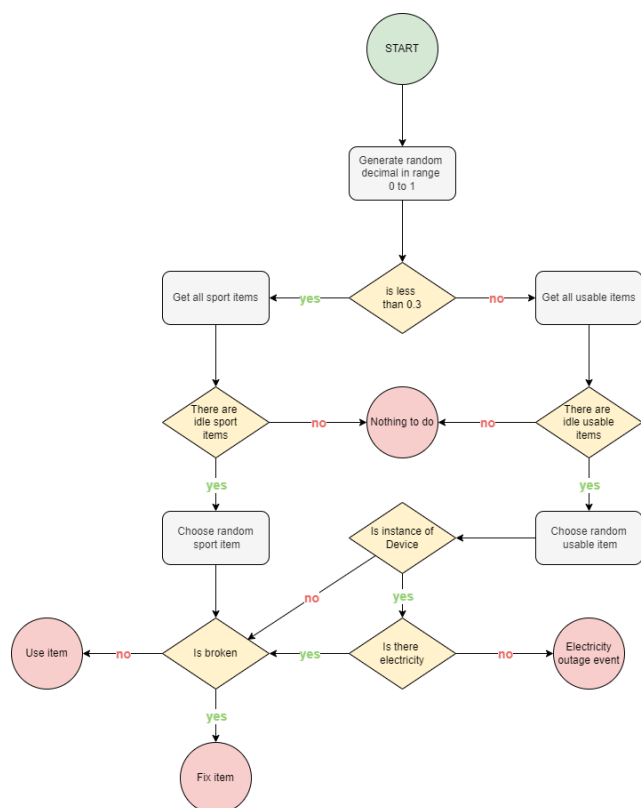
Pet action	Pet is busy	Pet age	Pet type
Play with toy	No	8 <= years < 20	Cat
Play with toy	Yes	5 <= years < 8	Hamster
Play with toy	No	0 <= years < 5	Turtle
Sleep	No	5 <= years < 8	Turtle
Sleep	No	years < 0	Dog
Sleep	Yes	years >= 20	Cat
Sleep	Yes	8 <= years < 20	Hamster
Eat home item	Yes	years < 0	Cat
Eat home item	No	years >= 20	Hamster
Eat home item	Yes	8 <= years < 20	Turtle
Eat home item	Yes	5 <= years < 8	Dog
Eat home item	No	0 <= years < 5	Dog
Play with toy	No	8 <= years < 20	Dog
Play with toy	Yes	0 <= years < 5	Cat

Play with toy	No	years < 0	Hamster
Play with toy	Yes	years >= 20	Turtle
Sleep	Yes	0 <= years < 5	Hamster
Sleep	Yes	years < 0	Turtle
Sleep	No	years >= 20	Dog
Sleep	Yes	8 <= years < 20	Dog
Sleep	No	5 <= years < 8	Cat

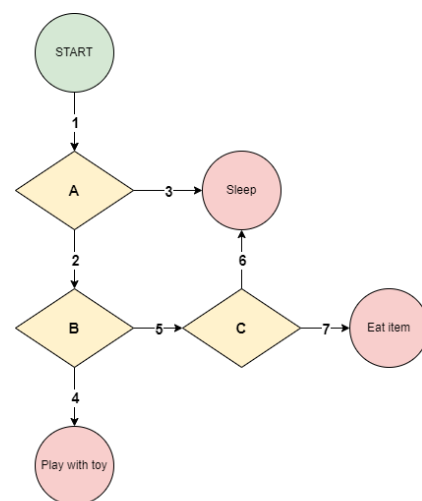
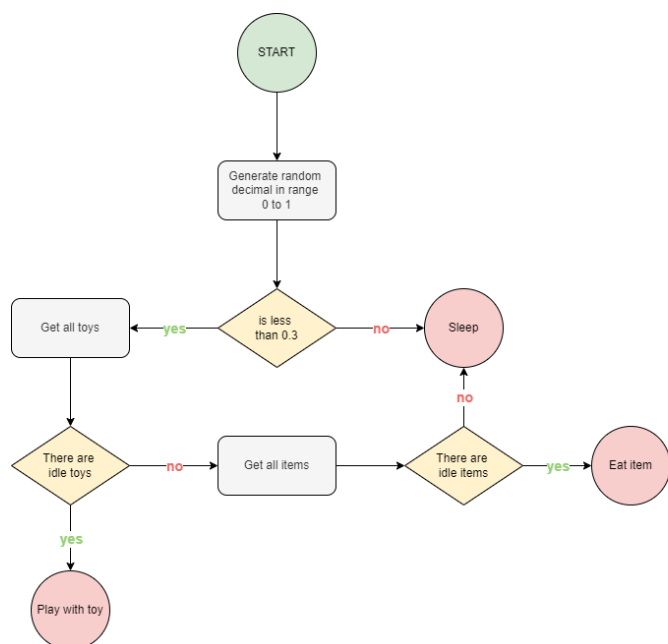
2.2. Testy průchodů

2.2.1. Vytvořte diagramy pro 2 procesy ve vaší aplikaci.

První diagram pro proces zodpovědný za výběr akce, kterou bude vykonávat osoba v aktuální iteraci:



Druhý diagram pro proces zodpovědný za výběr akce, kterou bude vykonávat domácí mazlíček v aktuální iteraci:



2.2.2. Vytvořte procesní testy z těchto diagramů s TDL 2.

Procesní testy z prvního diagramu:

Node	Sub-comb. of edges
A	1-2, 1-3
B	2-4, 2-6
C	3-5, 3-7, 3-8
D	6-9, 6-11, 7-9, 7-11, 10-9, 10-11, 12-9, 12-11
E	8-10, 8-13
F	13-12, 13-14

No.	Test sequence
1	1 – 2 – 4
2	1 – 3 – 5
3	1 – 2 – 6 – 9
4	1 – 3 – 7 – 9
5	1 – 3 – 8 – 10 – 9
6	1 – 3 – 8 – 13 – 12 – 9
7	1 – 2 – 6 – 11
8	1 – 3 – 7 – 11
9	1 – 3 – 8 – 10 – 11
10	1 – 3 – 8 – 13 – 12 – 11
11	1 – 3 – 8 – 13 – 14

Procesní testy z druhého diagramu:

Node	Sub-comb. of edges
A	1-2, 1-3
B	2-4, 2-5
C	5-6, 5-7

No.	Test sequence
1	1 – 2 – 5 – 6
2	1 – 3
3	1 – 2 – 5 – 7
4	1 – 2 – 4

2.3. Detailní testovací scénáře

2.3.1. Vytvořte 2 detailní testovací scénáře.

První testovací scénář z *PersonAndPetTest.java*:

Parametr	Obsah
ID testu	update_PersonIsUsingDevice_PetCanNotEatDevice
Název testu	Současné použití jednoho zařízení
Hloubka testu	Střední
Shrnutí testu	Domácí mazlíček nemůže kousat zařízení, pokud je právě v použití osobou.
Popis testu	<ol style="list-style-type: none"> Osoba a domácí mazlíček právě nemají žádnou aktivitu. Osoba začíná používat jediné volné zařízení. Domácí mazlíček se pokusí najít zajímavou aktivitu, ale žádná volná hračka ani zařízení v pokoji není. Domácí mazlíček usne. Osoba dokončí svou aktivitu a uvolní se.
Vstupní podmínky	<ul style="list-style-type: none"> > Pokoj, ve kterém je elektřina a pouze jedno zařízení > Jedna osoba a jeden domácí mazlíček v pokoji
Testovací data	<ul style="list-style-type: none"> > person.getCurrentAction(), person.isBusy() > pet.getCurrentAction(), pet.isBusy()
Očekávaný výsledek	pet.getCurrentAction() se rovná aktivitě "Pet wants to sleep".

Druhý testovací scénář z *PersonTest.java*:

Parametr	Obsah
ID testu	useSmtAtHome_usesBrokenItem_itemIsFixed
Název testu	Oprava rozbitého zařízení
Hloubka testu	Střední
Shrnutí testu	Osoba začne opravu, pokud vezme pro použití rozbité zařízení.
Popis testu	<ol style="list-style-type: none"> Osoba právě nemá žádnou aktivitu. Osoba začne používat jediné volné zařízení a zjistí, že je rozbité. Osoba přečte manuálové stránky pro opravu tohoto zařízení. Osoba opraví zařízení a uvolní se.
Vstupní podmínky	<ul style="list-style-type: none"> > Pokoj, ve kterém je elektřina a pouze jedno rozbité zařízení > Jedna osoba v pokoji
Testovací data	<ul style="list-style-type: none"> > person.getCurrentAction(), person.isBusy() > device.isBroken(), device.isBeingUsed()
Očekávaný výsledek	Device.isBroken() se rovná False.