

Projekt z předmětu SI 2 + SI 3

Modul pro mobilní zařízení

Návrh testů

Testování modulu

Tento dokument se zabývá testováním modulu pro mobilní zařízení, vyvíjeného v rámci projektu Restaurace FEL. Na následujících řádcích jsou uvedeny podmínky akceptačního testu, odpovídající funkčním požadavkům, kladeným na modul v analýze a dále informace a proběhlém white-box a black-box testování.

Návrh akceptačního testu

Zde následuje výpis jednotlivých testovacích případů tak, jak odpovídají funkčním požadavkům, a jejich navrhovaného řešení. Všechny testy by měly probíhat jednak white-box, tj. pomocí JUnit testů ve vývojovém prostředí Netbeans IDE, druhak black-box, tj. pomocí uživatelského rozhraní.

Test case : *Přidání nové objednávky*

Řešení : Dojde k pokusu o vytvoření nové objednávky a následnému vyhodnocení výsledků, objednávka by se měla shodovat s požadovaným vstupem.

Test case : *Úprava stávající objednávky*

Řešení : Dojde k pokusu o úpravu stávající objednávky a následnému vyhodnocení výsledků, objednávka by se měla shodovat s požadovaným vstupem.

Test case : *Zobrazení objednávek*

Řešení : Dojde k pokusu o vypsání všech objednávek příslušících účtu, výsledek bude poté porovnán s výstupem z databáze

Test case : *Zobrazování menu a jejich položek*

Řešení : Dojde k pokusu o vypsání všech menu, evidovaných v restauraci, a následně i jejich příslušných položek, výsledek bude poté porovnán s výstupem z databáze

Test case : *Založení nového účtu*

Řešení : Dojde k pokusu o vytvoření nového účtu s požadovanými parametry, tento bude následně (za předpokladu úspěšného vytvoření) porovnán s výstupem z databáze

Test case : *Úprava existujícího účtu*

Řešení : Dojde k pokusu o úpravu stávajícího účtu, změny by se měly v databázi projevit v souladu s očekáváním (tj. měly by být změněny právě ty hodnoty, které měly, zbytek beze změn)

Test case : *Přiřazení účtu osobě*

Řešení : Dojde k pokusu o přiřazení účtu osobě, tato změna by se měla v databázi projevit v souladu s očekáváním

Test case : *Přiřazené účtu stolu*

Řešení : Dojde k pokusu o přiřazení účtu stolu, tato změna by se měla v databázi projevit v souladu s očekáváním

Test case : *Přidání určité slevy k účtu*

Řešení : Dojde k pokusu o přiřazení slevy účtu, tato změna by se měla v databázi projevit v souladu s očekáváním

Test case : *Zobrazení všech účtů*

Řešení : Dojde k pokusu o vypsání všech zákaznických účtů, výsledek bude poté porovnán s výstupem z databáze

Test case : *Placení účtu*

Řešení : Dojde k pokusu o zaplacení účtu, a to jak jeho části, tak i účtu celého. Výsledek by se měl v databázi projevit očekávaným způsobem (označení položek a případně i celého účtu jako zaplaceného)

Test case : *Administrace*

Řešení : Dojde k pokusu o změnu připojovacích nastavení a následně k otestování, zda změna skutečně proběhla a nová konfigurace byla uložena.

Test case : *Omezenost přístupu*

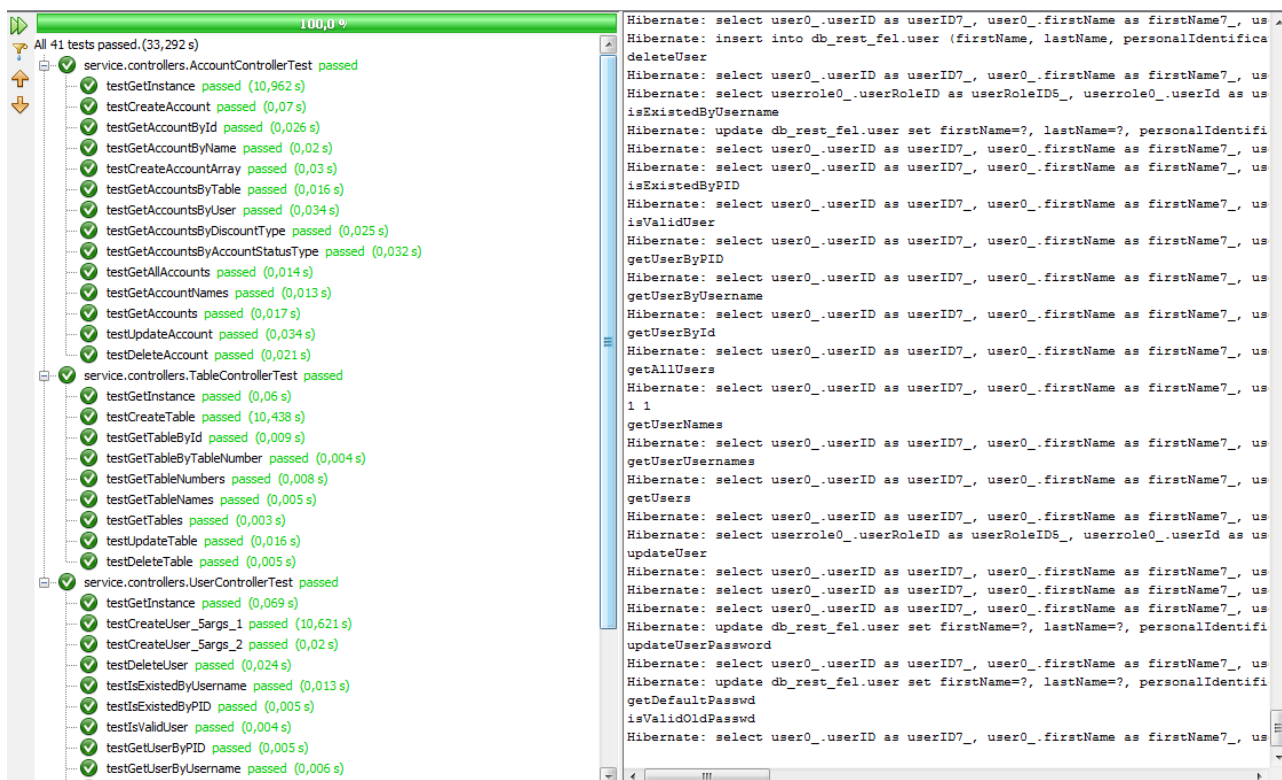
Řešení : Dojde k pokusu o nepovolené přihlášení, které by mělo být odmítnuto, a následně o přihlášení povolené, které by mělo uživateli umožnit vstup do systému

White-box testování

White-box testování bude probíhat „zvnitřku“ aplikace, tj. bude využívat a testovat zejména business logiku a to pomocí JUnit testů. Při testování se rovněž počítá s využitím pluginu pro Netbeans IDE, který umožňuje sledovat pokrytí kódu aplikace testy.

Na obr. 1 níže je vidět ukázka unit testů, spuštěných v Netbeans. Tento ukázkový test byl vytvořen tak, aby fungoval s modelovou (v tomto případě prázdnou) databází a v průběhu testů postupně otestoval funkčnost veškerých metod, týkajících se osob, účtů a stolů, které se starají o přístup k databázi. Jak je z obrázku vidět, testy dopadly úspěšně.

Na obr. 2 je možné prohlédnout si ukázku pokrytí kódu pomocí výše zmiňovaného pluginu pro Netbeans. Zelené řádky značí kompletně pokryté, oranžové pak řádky, které byly pokryté jen zčásti (např. protože se na řádku nachází podmínka, která buď nebyla vyhodnocena celá nebo nebyla splněna).



Obr. 1 – Ukázka proběhnuvších unit testů



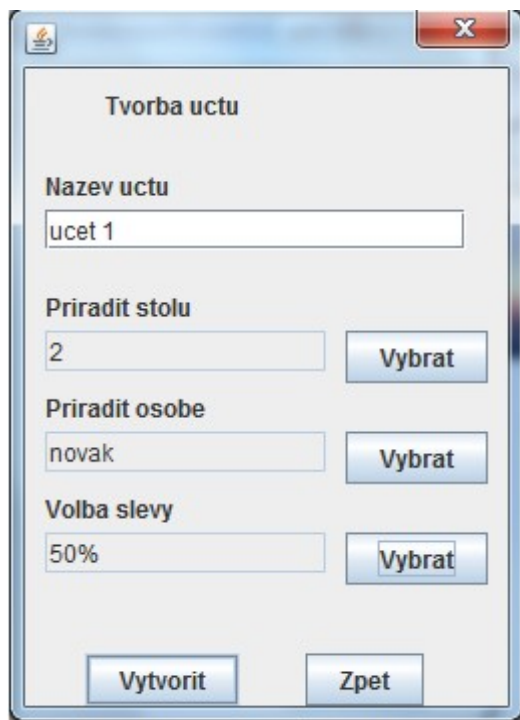
Obr. 2 – Ukázka pokrytí kódu ve třídě AccountController

Black-box testování

Black-box testování probíhalo jednak pomocí několika nedobrovolných dobrovolníků z řad laiků, druhak pomocí frameworku Abbot + Costello, který umožňuje automaticky testovat grafické rozhraní aplikace a snadno odhalit případné nefungující prvky. Ná následujících obrázcích

jsou vidět ukázky z běhu aplikace. Na obr. 3 je to tvorba nového účtu a na obr. 4 změna nastavení aplikace.

Při práci s frameworkem docházelo k několika problémům, např. v okamžiku, kdy došlo k přihlášení, se celá aplikace zavřela místo toho, aby se otevřel nový panel (i když při standardním spuštění aplikace tento případ nikdy nenastal) aj. Nakonec se je však snad povedlo uspokojivě vyřešit a testy dopadly úspěšně.



Tvorba účtu

Název účtu
ucet 1

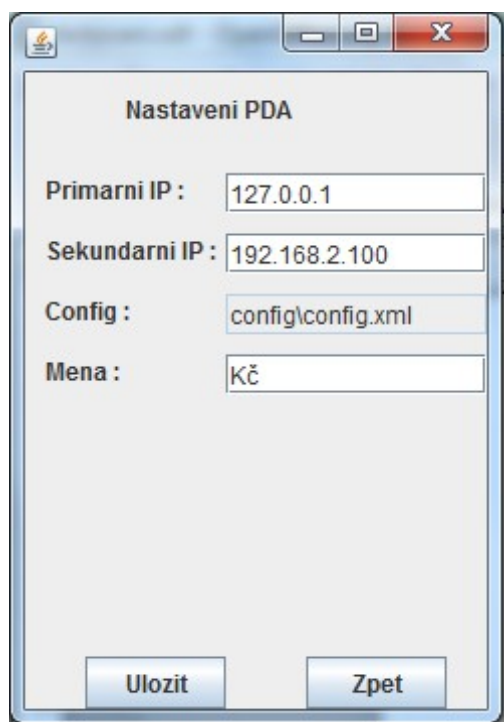
Priradit stolu
2 Vybrat

Priradit osobe
novak Vybrat

Volba slevy
50% Vybrat

Vytvorit Zpet

Obr. 3 – Vytváření účtu na mobilním zařízení



Nastavení PDA

Primární IP : 127.0.0.1

Sekundární IP : 192.168.2.100

Config : config\\config.xml

Mena : Kč

Uložit Zpet

Obr. 4 – Ukazka změny nastavení v mobilním zařízení