# 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, druhnak black-box, tj. pomocí uživateslkého rozhraní.

Test case: Přídání 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 pokus o vypsání všech objednávek přislušících 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 učtu s požadovanými parametry, tento bude následně (za předpokladu úspěšného vytvoření) porovnán s výstupem s 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: Přiřazeni 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: Přídání 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 datábazi projevit očekáváným 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 bussiness 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).

```
Hibernate: select user0 .userID as userID7 , user0 .firstName as firstName7 ,
All 41 tests passed.(33,292 s)
                                                                                        Hibernate: insert into db_rest_fel.user (firstName, lastName, personalIdentifica
                                                                                        deleteUser
   service.controllers.AccountControllerTest
                                                                                        Hibernate: select user0_.userID as userID7_, user0_.firstName as firstName7_,
         testGetInstance passed (10,962 s)
                                                                                        Hibernate: select userrole0 .userRoleID as userRoleID5 , userrole0 .userId as us
                                                                                        isExistedByUsername
         testGetAccountById passed (0,026 s)
                                                                                        Hibernate: update db_rest_fel.user set firstName=?, lastName=?, personalIdentifi
                                                                                       Hibernate: select user0_.userID as userID7_, user0_.firstName as firstName7_, us
Hibernate: select user0_.userID as userID7_, user0_.firstName as firstName7_, us

    testGetAccountBvName passed (0.02 s)

    testCreateAccountArray passed (0,03 s)

                                                                                        isExistedBvPID

▼ testGetAccountsByTable passed (0,016 s)

                                                                                                    select userO .userID as userID7 , userO .firstName as firstName7 , us
         testGetAccountsByUser passed (0,034s)
                                                                                        isValidUser
         testGetAccountsByDiscountType passed (0,025 s)
                                                                                                   select user0_.userID as userID7_, user0_.firstName as firstName7_, us

    testGetAccountsByAccountStatusType passed (0,032 s)

                                                                                        getUserByPID
         testGetAllAccounts passed (0,014s)
                                                                                        Hibernate: select user0_.userID as userID7_, user0_.firstName as firstName7_, us
         testGetAccountNames passed (0,013 s)
                                                                                        getUserByUsername
         testGetAccounts passed (0,017 s)
                                                                                        Hibernate: select user0_.userID as userID7_, user0_.firstName as firstName7_, us

    testUpdateAccount passed (0,034s)

                                                                                        getUserById
         testDeleteAccount passed (0,021s)
                                                                                        Hibernate: select user0 .userID as userID7 , user0 .firstName as firstName7 , us
      service.controllers.TableControllerTest page
                                                                                        Hibernate: select user0_.userID as userID7_, user0_.firstName as firstName7_, us
         testGetInstance passed (0,06 s)

    testCreateTable passed (10,438 s)

                                                                                        getUserNames
         testGetTableById passed (0,009 s)
                                                                                        Hibernate: select user0_.userID as userID7_, user0_.firstName as firstName7_, us
         testGetTableByTableNumber passed (0,004s)
                                                                                        getUserUsernames
         testGetTableNumbers passed (0,008 s)
                                                                                        Hibernate: select user0_.userID as userID7_, user0_.firstName as firstName7_, us

    testGetTableNames passed (0.005 s).

                                                                                        getUsers
                                                                                        -
Hibernate: select user0_.userID as userID7_, user0_.firstName as firstName7_, us
         testGetTables passed (0,003 s)
         testUpdateTable passed (0,016 s)
                                                                                        Hibernate: select userrole0_.userRoleID as userRoleID5_, userrole0_.userId as us
                                                                                        updateUser
         testDeleteTable passed (0,005 s)
                                                                                        ...
Hibernate: select user0_.userID as userID7_, user0_.firstName as firstName7_, us
   service.controllers.UserControllerTest passed
                                                                                       Hibernate: select user0_.userID as userID7_, user0_.firstName as firstName7_, us
Hibernate: select user0_.userID as userID7_, user0_.firstName as firstName7_, us
         testGetInstance passed (0,069 s)
         testCreateUser_5args_1 passed (10,621 s)
                                                                                       Hibernate: update db_rest_fel.user set firstName=?, lastName=?, personalIdentifi

▼ testCreateUser_5args_2 passed (0,02 s)

         testDeleteUser passed (0,024s)
                                                                                        Hibernate: select userO .userID as userID7 , userO .firstName as firstName7 , us
                                                                                        Hibernate: update db_rest_fel.user set firstName=?, lastName=?, personalIdentifi
         testIsExistedByUsername passed (0,013 s)
                                                                                        getDefaultPasswd
         testIsExistedByPID passed (0,005 s)
                                                                                        isValidOldPasswd
         testIsValidUser passed (0,004s)
                                                                                        Hibernate: select user0_.userID as userID7_, user0_.firstName as firstName7_, us
         testGetUserByPID passed (0,005 s)
         testGetUserByUsername passed (0,006 s)
```

Obr. 1 – Ukázka proběhnuvších unit testů

```
private AccountController() {
public static AccountController getInstance() {
   if (instance == null) {
        instance = new AccountController();
                                                                                                    Ξ
    return instance;
//vvtvari zaznam o novem uctu
public boolean createAccount(String name, int accountStatusTypeId, int tableId, int userId, int di
   if (name!=null&&!name.equals("")) {
       Account a = getAccountByName(name);
        if (a!=null&&a.getName().eguals(name)/*&&a.getAccountStatusType().getAccountStatusTypeId()
        account = new Account();
       account.setName(name);
       AccountStatusType accountStatusType = AccountStatusType.findById(accountStatusTypeId);
        account.setAccountStatusType(accountStatusType);
       Table table = Table.findById(tableId);
        account.setTable(table);
        User user = User.findById(userId);
        account.setUser(user);
        DiscountType discountType = DiscountType.findById(discountTypeId);
       account.setDiscountType(discountType);
        account.create();
```

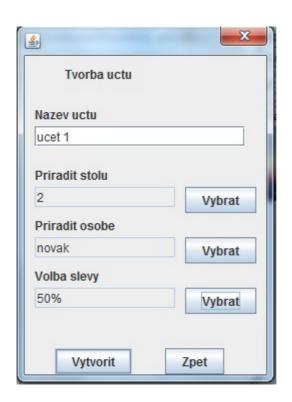
Obr. 2 – Ukázka pokrytí kódu ve třídě AccountController

### Black-box testování

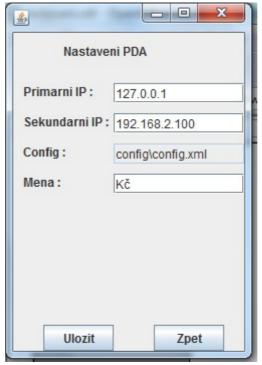
Black-box testování probíhalo jednak pomocí několika nedobrovolných dobrovolníků z řad laiků, druhnak pomocí frameworku Abbot + Costello, který umožňuje automaticky testovat grafické rozhraní aplikace a snadno odhalit případné nefungující prvky. Ná následujících obrázcích

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.



Obr. 3 – Vytvareni uctu na mobilnim zarizeni



*Obr.* 4 – *Ukazka zmeny nastaveni v mobilnim zarizeni*