

České vysoké učení technické

Fakulta elektrotechnická

Plán testů

Systém elektronických voleb

E-volby

Obsah

Úvod.....	3
Jednotkové testování.....	3
Integrační testování.....	3
Přístup k integračnímu testování.....	3
Integrační testování projektu.....	3
Validační testování.....	4
Použití v projektu.....	4
Testování webového rozhraní.....	4
Systémové testování.....	4
Výsledky testů.....	5

Úvod

Účelem testování je odhalení chyb vzniklých při programování, chování aplikace v nestandardních situacích a zajištění bezproblémového chodu aplikace v jednotlivých uživatelských scénářích. V plánu testování projektu e-volby je využíváno především *White-box* a *Black-box* testování.

Tento dokument definuje přístupy k testování systému E-volby, použité nástroje a očekávané výsledky.

Jednotkové testování

Jednotkové testy fungují na principu rozdělení aplikace na malé části a testování těchto částí bez závislosti na ostatních, s předpokladem znalosti implementace.

White-box testování v projektu e-volby zajišťují jednotkové testy, konkrétně je to testovací framework JUnit, integrovaný do vývojového prostředí NetBeans.

Jednotkových testů bude v systému použito pro testování částí kódu psaných v obyčejné javě.

Integrační testování

Přístup k integračnímu testování

V projektu E-volby bude pro integrační testování použit přístup "velkého třesku". Nejdříve se spojí všechny moduly a pak je testováno vše naráz. Projekt není rozsáhlý, proto je možné tento přístup použít.

Integrační testování projektu

Integrační testování se zaměřuje na vlastní přístup k datům a komunikaci. V případě projektu E-volby jsou to hlavně EJB (*Enterprise JavaBean*) moduly. Automatické testování EJB modulů je v projektu e-volby obtížné, protože na automatické testování nebylo myšleno při zahájení vývoje. Testování tedy bude zajištěno manuálně, přímou kontrolou obsahu tabulek v databázi po provedení uživatelských akcí. Tato metoda je jednoduchá, ale bude časově náročná. Je to však efektivnější a méně náročné řešení než přepracování projektu pro umožnění automatizovaného testování a samotné vytváření automatických testů.

Touto kontrolou se otestuje správná komunikace jednotlivých modulů i jejich komunikace pomocí webového rozhraní.

Další kapitolu tvoří volební applet. Po zasazení nového appletu je nutné otestovat správnost předávaných parametrů na vstup a správnost výsledků.

Validační testování

Použití v projektu

Validační testování na projekt nahlíží z pohledu zákazníka jako na *black-box* se skrytou implementací.

Black-box testování je v projektu e-volby použito jako testování celkové funkčnosti projektu, tedy funkčnosti webové uživatelského rozhraní.

Testování webového rozhraní

Testování webového rozhraní je zajištěno pomocí testovacího frameworku Selenium. Tento framework umožňuje nahrát skripty uživatelských akcí, jejich automatické či poloautomatické provedení a kontrolu správnosti provedení.

Testerovi skript umožňuje rychlou a spolehlivou kontrolu, zda po aplikaci funguje z uživatelského pohledu tak, jak by měla.

Webové rozhraní bude otestováno pro všechny standardní uživatelské akce.

Systémové testování

Systémové testování tvoří závěrečnou část testování a ověřuje celkovou funkčnost aplikace v simulovaném prostředí a při reálném nasazení.

Zátěžový test bude proveden monitorováním systémových prostředků (zatížení procesoru, paměti, síťové komunikace a potřeba místa na disku) pomocí simulování nestandardních situací s testovacími daty a poté při běžném provozu.

Testování obnovy se týká uložení dat. Je třeba otestovat plánované zálohy dat z databáze, funkčnost vrácení databáze do stavu před chybou a schopnost obnovit data po restartu či pádu databáze a aplikace. Testování obnovy se bude týkat reálně nasazeného databázového stroje a práce s ním. V současné době je zvolen databázový systém Mysql. Předmětem testování musí být prověření schopností zaškolené obsluhy serveru tyto úkony zvládnout.

Bezpečnostní testování musí prověřit zabezpečení systému před útoky zvenčí a manipulaci s daty. Projekt elektronických voleb je záležitost, která musí být naprosto bezpečná a je třeba zaručit, že data udržovaná v aplikaci nemohou být upravována. Bezpečnostní testování bude provedeno zkouškou získání neautorizovaného přístupu k datům přes samotnou aplikaci nebo přímo fyzický stroj. Testování bude zajištěno z pohledu vývojářů i nezávislých osob, které zkusí do systému proniknout a zfalšovat volby.

Výsledky testů

Výstupy z testování budou obsahovat informace o provedení, obsahu a výstupu daného testu. V případě neúspěšného testu musí výsledek obsahovat důvod neúspěchu a možné kroky k odstranění problému s neúspěšným testem.