

**Management projektů**

**Systém pro správu rizik**

**StackBoss**

**Testy**

| Historie | | | | |
| --- | --- | --- | --- | --- |
| Verze | Datum | Status | Kdo | Poznámka |
| 1.0 | 5.5. |  | Daniel Kamenický |  |
|  |  |  |  |  |

1 Úvod 2

2 Postup testování 2

3 Testy přírůstku 2

4 Integrační testy 3

5 Uživatelské testy 3

5.1 Operace nad projekty 3

5.1.1 Testovací scénář 1 – Projects\_GetAll 3

5.1.2 Testovací scénář 2 – InsertProject 3

5.1.3 Testovací scénář 3 – GetProject 3

5.1.4 Testovací scénář 4 – UpdateProject 3

5.1.5 Testovací scénář 5 – DeleteProject 3

5.2 Operace nad riziky 3

5.2.1 Testovací scénář 1 – Risks\_GetAll 3

5.2.2 Testovací scénář 2 – InsertRisk 3

5.2.3 Testovací scénář 3 – GetRisk 3

5.2.4 Testovací scénář 4 – UpdateRisk 3

5.2.5 Testovací scénář 5 – DeleteRisk 3

# Úvod

Tento dokument definuje testování výsledného produktu Systém pro správu rizik. V dokumentu jsou zběžně popsány testy, které byly provedeny.

# Postup testování

Pro testování byl vytvořen samostatný modul „StackbossDbContextTests.cs“, ve kterém je implementace jednotlivých testů. Pro testy nebyl využit žádný přídavný nástroj. Jednotlivé testy byly vytvořeny testerem. Každý test obsahuje předem definovanou hodnotu (očekávaný výstup), s kterou se porovnává výsledek jednotlivé metody.

Kritéria dokončení testů jsou následovné:

* Všechny vytvořené testy musí skončit úspěchem
* Všechny testovací scénáře produkují očekávaný výstup
* Jsou otestovány všechny požadované části systému

# Testy přírůstku

V průběhu vývoje byla po celou dobu testovány jednotlivé části (nová funkcionalita), která byla přidána. Testování probíhalo pomocí takzvaných „unit testů“ a pokud nebyl unit test vytvořen, tak přidaná funkcionalita byla otestována ručně vývojářem.

Funkcionalita byla ručně testována na jednotlivých online setkání, kde byly vyhrazeny připomínky při špatném/nepředpokládaném chování a poté vývojářem upraveny.

# Integrační testy

Integrační testování bylo provedeno testovacím týmem po dokončení testů přírůstku vývojářem. Integrační testování kontrolovalo bezchybnou komunikaci jednotlivých vnitřních komponent produktu. Testování bylo prováděno manuálně a případné chyby byly konzultovány a opraveny vývojářem.

# Uživatelské testy

Uživatelské testy jsou definovány v dokumentu návrh. Dělí se na dvě základní části, a to testování operací nad projekty. Tyto testy kontrolují správný proces vytváření, aktualizování, mazání a získávání jednotlivých projektů. Ve zdrojovém kódu jsou metody, které pracují s databází, přesněji s databázovou tabulkou uchovávající záznamy o projektech. Každá metoda představuje jednu CRUD operaci a ta představuje jeden use-case testovacího scénáře. Druhou částí jsou operace nad riziky. Tyto testy kontrolují správný proces vytváření, aktualizování, mazání a získávání jednotlivých rizik. Implementace je podobná jak u projektů, avšak metody operují nad databázovou tabulkou uchovávající záznamy jednotlivých rizik. Podobně, jak u projektů, bude jedna metoda implementovat jednu z operací, a tedy bude představovat jeden use-case testovacího scénáře.

## Operace nad projekty

### Testovací scénář 1 – Projects\_GetAll

Identifikátor případu: Uživatel si chce zobrazit počet všech definovaných projektů.

Název testovacího modulu: GetAllProjectsAsync

Účel testoacího modulu: Získání počtu všech definovaných projektů.

| Datum | Čas | Tester | Stav | Výstupní hodnota |
| --- | --- | --- | --- | --- |
| 3.5. | 10:00 | Damián Gladiš | O.K. | Int: 4 |

### Testovací scénář 2 – InsertProject

Identifikátor případu: Uživatel si chce zadefinovat nový projekt.

Název testovacího modulu: InsertProjectAsync

Účel testoacího modulu: Vložení nového projektu.

| Datum | Čas | Tester | Stav | Výstupní hodnota |
| --- | --- | --- | --- | --- |
| 3.5. | 10:30 | Damián Gladiš | O.K. | Bool: True |

### Testovací scénář 3 – GetProject

Identifikátor případu: Uživatel si chce otevřít již definovaný projekt.

Název testovacího modulu: GetProjectAsync

Účel testoacího modulu: Získání konkrétního projektu.

| Datum | Čas | Tester | Stav | Výstupní hodnota |
| --- | --- | --- | --- | --- |
| 3.5. | 11:30 | Damián Gladiš | O.K. | String: “Medical IS” |

### Testovací scénář 4 – UpdateProject

Identifikátor případu: Uživatel chce provést změnu v již definovaném projektu.

Název testovacího modulu: UpdateProjectAsync

Účel testoacího modulu: Aktualizace již vytvořeného projektu.

| Datum | Čas | Tester | Stav | Výstupní hodnota |
| --- | --- | --- | --- | --- |
| 3.5. | 11:45 | Damián Gladiš | O.K. | Bool: True |

### Testovací scénář 5 – DeleteProject

Identifikátor případu: Uživatel chce smazat projekt.

Název testovacího modulu: DeleteProjectAsync

Účel testoacího modulu: Smazání projektu.

| Datum | Čas | Tester | Stav | Výstupní hodnota |
| --- | --- | --- | --- | --- |
| 3.5. | 13:00 | Damián Gladiš | O.K. | Bool: True |

## Operace nad riziky

### Testovací scénář 1 – Risks\_GetAll

Identifikátor případu: Uživatel si chce zobrazit počet všech definovaných rizik.

Název testovacího modulu: GetAllRisksAsync

Účel testoacího modulu: Získání počtu všech definovaných rizik.

| Datum | Čas | Tester | Stav | Výstupní hodnota |
| --- | --- | --- | --- | --- |
| 3.5. | 14:00 | Damián Gladiš | O.K. | Int: 18 |

### Testovací scénář 2 – InsertRisk

Identifikátor případu: Uživatel chce vložit nové riziko.

Název testovacího modulu: InsertRiskAsync

Účel testoacího modulu: Vložení nového rizika.

| Datum | Čas | Tester | Stav | Výstupní hodnota |
| --- | --- | --- | --- | --- |
| 3.5. | 14:30 | Damián Gladiš | O.K. | Bool: True |

### Testovací scénář 3 – GetRisk

Identifikátor případu: Uživatel si chce zobrazit již definované riziko.

Název testovacího modulu: GetRiskAsync

Účel testoacího modulu: Získání konkrétního rizika.

| Datum | Čas | Tester | Stav | Výstupní hodnota |
| --- | --- | --- | --- | --- |
| 3.5. | 15:00 | Damián Gladiš | O.K. | String: “Risk of Losing customers” |

### Testovací scénář 4 – UpdateRisk

Identifikátor případu: Uživatel chce upravit již definované riziko.

Název testovacího modulu: UpdateRiskAsync

Účel testoacího modulu: Aktualizace již vytvořeného rizika.

| Datum | Čas | Tester | Stav | Výstupní hodnota |
| --- | --- | --- | --- | --- |
| 3.5. | 15:15 | Damián Gladiš | O.K. | Bool: True |

### Testovací scénář 5 – DeleteRisk

Identifikátor případu: Uživatel chce smazat riziko.

Název testovacího modulu: DeleteRiskAsync

Účel testovacího modulu: Smazání rizika.

| Datum | Čas | Tester | Stav | Výstupní hodnota |
| --- | --- | --- | --- | --- |
| 3.5. | 15:45 | Damián Gladiš | O.K. | Bool: True |