Pflichtenheft – SmartQS

Birngruber, Werner

|  |  |
| --- | --- |
| **Projektbezeichnung** | SmartQS |
| **Projektleiter** | Lukas Werner |
| **Erstellt am** | 25.09.2020 |
| **Letzte Änderung am** | 25.09.2020 |
| **Status** | In Bearbeitung |
| **Aktuelle Version** | 1.1 |

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Änderungen | Datum | Autor |
| 1.0 | Pflichtenheft erstellt | 25.09.2020 | Birngruber, Werner |
| 1.1 | Name finalisiert, Bilder eingefügt | 02.10.2020 | Birngruber |

Inhalt

[1 Einleitung 3](#_Toc52527113)

[2 Allgemeines 3](#_Toc52527114)

[2.1 Ausgangssituation 3](#_Toc52527115)

[2.2 Teams und Schnittstellen 3](#_Toc52527116)

[3 Konzept 3](#_Toc52527117)

[3.1 Ziel des Anbieters 3](#_Toc52527118)

[3.2 Ziele und Nutzen des Anwenders 3](#_Toc52527119)

[4 Funktionale Anforderungen 3](#_Toc52527120)

[4.1 Integrierbare Lösung in MS Teams (als Tab) 3](#_Toc52527121)

[4.2 Plans & Runs definieren 3](#_Toc52527122)

[4.3 Runs durchführen und Ergebnisse Dokumentieren 3](#_Toc52527123)

[4.4 Ablage in SharePoint/Team und Versand einer „Card“ 3](#_Toc52527124)

[4.5 Default Tests integrieren 3](#_Toc52527125)

[4.6 Speicherung in Azure SQL und Zugriff über mongoose 3](#_Toc52527126)

[5 Nichtfunktionale Anforderungen 3](#_Toc52527127)

[5.1 UI so einfach wie möglich 3](#_Toc52527128)

[6 Ziele 3](#_Toc52527129)

[6.1 Muss-Ziele 3](#_Toc52527130)

[6.2 Soll-Ziele 4](#_Toc52527131)

[6.3 Wunsch-Ziele 4](#_Toc52527132)

[7 Rahmenbedingungen 4](#_Toc52527133)

[7.1 Meilensteinplan 4](#_Toc52527134)

[7.2 Technische Anforderungen 4](#_Toc52527135)

[7.3 Verwendetet Technologien 4](#_Toc52527136)

[8 Use Case Diagramm 5](#_Toc52527137)

[9 GUI Entwurf 5](#_Toc52527138)

[10 Abnahmebedingungen 6](#_Toc52527139)

# 1 Einleitung

Dieses Pflichtenheft alle funktionalen und nicht funktionalen Anforderungen für das Tool zum manuellen Software-Testing. Somit werden die Rahmenbedingungen für die Entwicklung festgelegt.

# 2 Allgemeines

## 2.1 Ausgangssituation

Zurzeit wird im Unternehmen zum manuellen Software-Testing noch kein Tool verwendet. Um das Software-Testing zu erleichtern, soll deshalb ein Tool entwickelt werden, welches direkt über Microsoft Teams verwendet werden kann.

## 2.2 Teams und Schnittstellen

|  |  |  |  |
| --- | --- | --- | --- |
| Rolle(n) | Aufgabenbereich | Name | E-Mail |
| Projektleiter/Entwickler | Frontend/Design | Lukas Werner | 40146720160360@litec.ac.at |
| Entwickler | Backend/Schnittstellen | Dominik Birngruber | 40146720160258@litec.ac.at |

# 3 Konzept

## 3.1 Ziel des Anbieters

Eine voll funktionsfähige App die in Microsoft Teams als Tab eingefügt werden kann und alle Daten auf SharePoint speichert.

## 3.2 Ziele und Nutzen des Anwenders

Eine App für Microsoft Teams in der manuelle Software-Tests definiert werden können. Diese sollten anschließend durchgeführt werden können und e soll automatisch eine Dokumentation erstellt werden und auf SharePoint gespeichert werden. Danach soll eine „Card“ in Teams versendet werden. Die Daten werden in Azure SQL gespeichert.

# 4 Funktionale Anforderungen

## 4.1 Integrierbare Lösung in MS Teams (als Tab)

## 4.2 Plans & Runs definieren

## 4.3 Runs durchführen und Ergebnisse Dokumentieren

## 4.4 Ablage in SharePoint/Team und Versand einer „Card“

## 4.5 Default Tests integrieren

## 4.6 Speicherung in Azure SQL und Zugriff über mongoose

# 5 Nichtfunktionale Anforderungen

## 5.1 UI so einfach wie möglich

# 6 Ziele

## 6.1 Muss-Ziele

* Plans und Runs definieren
* Runs durchführen
* Ergebnisse auf SharePoint speichern
* Speicherung der Daten in CosmosDB auf der Azure Plattform
* Versand einer „Card“ in Teams

## 6.2 Soll-Ziele

* Integrierbare Lösung in Teams
* Default Tests integrieren

## 6.3 Wunsch-Ziele

* Schnittstellen Anbindung Azure DevOps
* Generierung eines „Change-Log“ E-Mails

# 7 Rahmenbedingungen

## 7.1 Meilensteinplan

* alle User Stories definiert 15.10.2020
* Design - Mockup fertiggestellt 31.10.2020
* erste Beta zum Testen freigegeben 27.12.2020
* erste public Beta freigegeben 25.02.2021
* Dokumentation fertiggestellt 07.03.2021

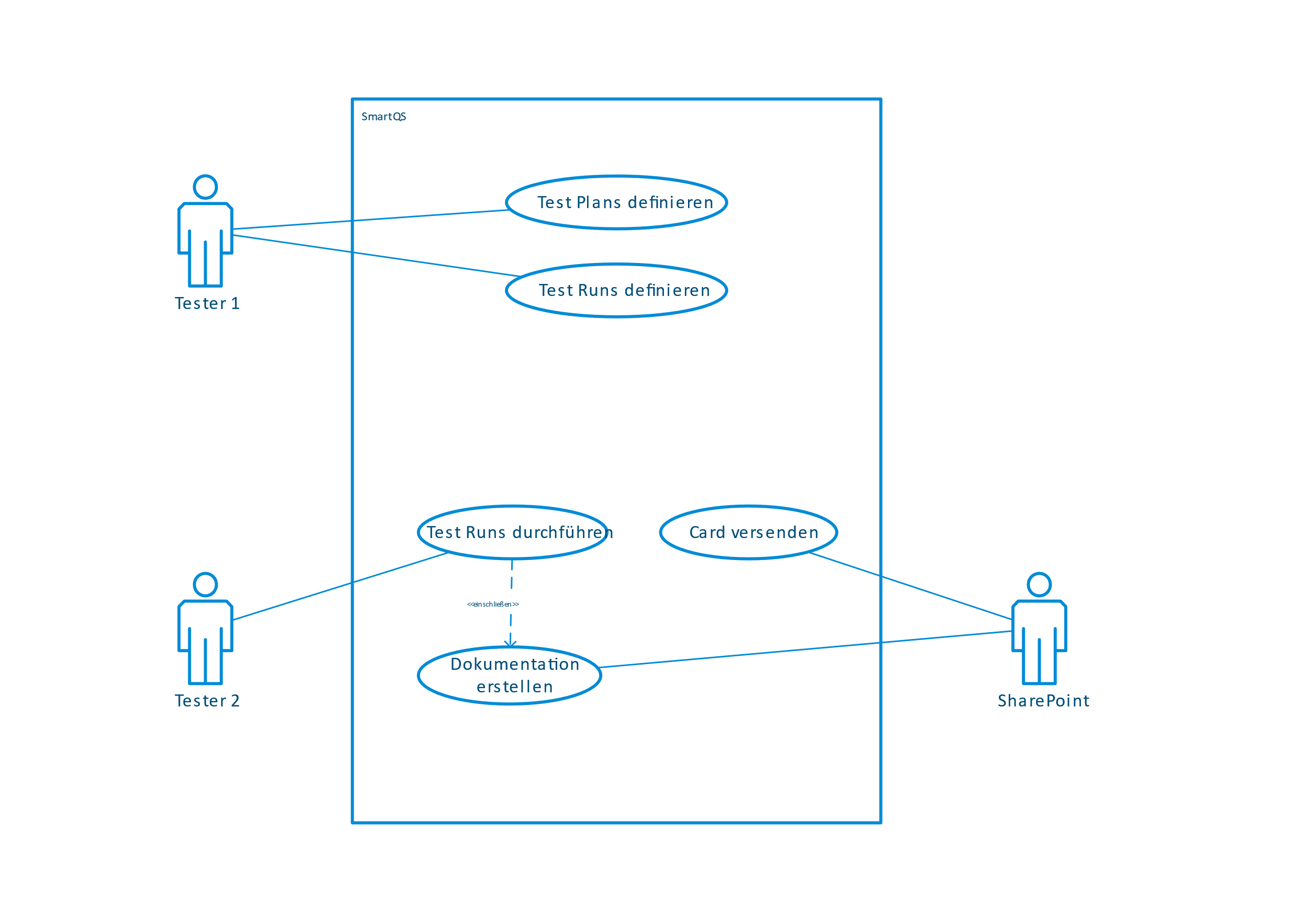
## 7.2 Technische Anforderungen

* SharePoint
* Mongo DB auf Azure

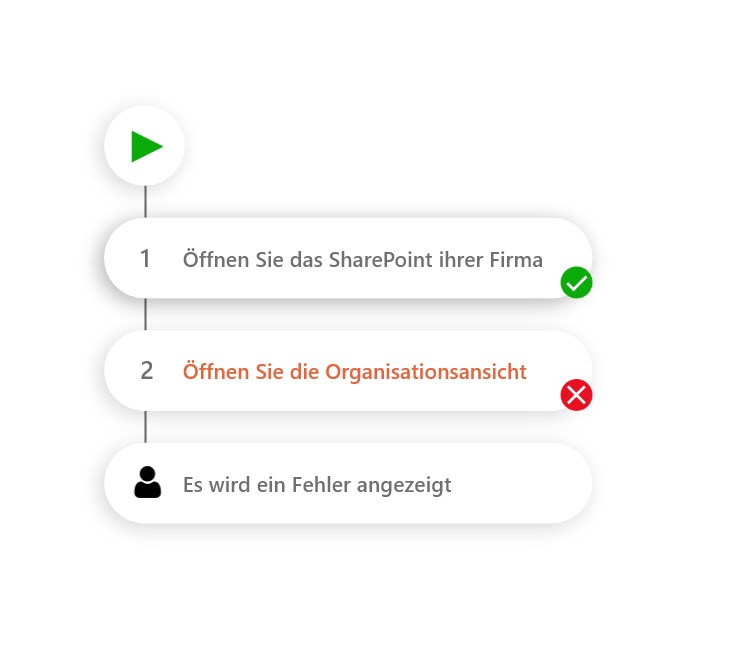
## 7.3 Verwendetet Technologien

* CosmosDB
* React
* Mongoose

# 8 Use Case Diagramm



# 9 GUI Entwurf



# 10 Abnahmebedingungen

* Getestetes Tool für Software-Testing
* Installationsanleitung