

# Software Requirements Specification

## 1. Introduction

### 1.1 Purpose

The purpose of the project is to create a Q&A platform, where users can ask and answer questions, view existing answers and communicate with each other on different topics. StarForum is a place for gaining knowledge as well as sharing it.

### 1.2 Intended Audience

Users of the software are students who need help and ask questions for their studies; students, who have required knowledge and want to help by giving answers to asked questions; people, who start learning new topics and need support.

## 2. System Features and Requirements

### 2.1 Functional Requirements

Were previously described and can be found in the [following](#) document.

### 2.2 External Interface Requirements

External Interfaces include Google, which is used for authentication and authorization.

Frontend stack: Angular, Typescript.

Backend stack: .NET, C#, Entity Framework, MS SQL.

Azure, which is used for storage and networking.

Hardware interfaces: both Mac and Windows operating systems through their default web browser.

### 2.3 System Features

Some of the main system features are described below.

### 2.3.1 Creation of question

#### 2.3.1.1 Description And Priority

Users can create a question by providing Summary, Description and specifying Tags. Only users that are signed into the system can access this feature.

Priority: high.

#### 2.3.1.2 Response Sequences

Action - user presses 'Create a question' button. Response - modal for creating a question is opened.

Action - user adds tags to entering the word and presses 'Enter' key.

Result - tags are added to the field and displayed as separate words.

Action - user does not fill all required fields and presses 'Enter'. Result - empty fields are highlighted in red.

Action - user fills all required fields and presses 'Enter'. Result - question is created.

### 2.3.2 Adding to favorites

#### 2.3.2.1 Description And Priority

Users should have a possibility to add questions to favorites, remove from favorites and view list of questions, which were added to favorites. By having favorite questions displayed on a separate tab, users are able to quickly find important items.

Priority: high.

#### 2.3.2.2 Response Sequences

Action - user clicks on unfilled star on question item next to question title. Response - question is added to 'Favorites', star is changed to filled.

Action - user clicks on filled star on question item next to question title. Response - question is removed from 'Favorites', star is changed to unfilled.

Action - user clicks on 'Favorites' page. Response - the page with questions marked as favorites is displayed.

Action - user clicks on filled star on question item on 'Favorites' page. Response - question is removed from favorites, page is refreshed.

### 2.3.3 Adding a complaint

#### 2.3.3.1 Description And Priority

Users should have a possibility to add a complaint about users and/or their answers and questions in order for administrators to receive complaints and work them through.

Priority: medium.

#### 2.3.3.2 Response Sequences

Action - user clicks on 'Complain' button next to answer under the question item. Response - dropdown with complaint reasons is displayed.

Action - user selects one of the displayed reasons. Response - message about successful complaint is displayed; request about blocking the author of the answer is sent to the administrator.

Action - user with administrator rights opens 'Complaints' page. Response - list of complaints is displayed.

Action - user with administrator rights clicks on 'Block' option on complaint. Response - user is blocked.

#### 2.4 Safety Requirements

Databases should use sharding to be redundant to prevent loss of data. Backups of the databases should be done hourly and be kept for one week.

#### 2.5 Security Requirements

Any keys used for the REST API should be stored securely.

Only the REST API should be able to connect to the databases.

Databases should be behind a firewall.