# **AskAMech Test plan**

Table of Contents

[**AskAMech Test plan** 1](#_Toc33018693)

[**Introduction** 2](#_Toc33018694)

[1. Objectives 2](#_Toc33018695)

[1.1 Purpose 2](#_Toc33018696)

[1.2 Scope and level of testing 2](#_Toc33018697)

[1.2.1 Features to be tested 3](#_Toc33018698)

[1.2.2 Features not to be tested 3](#_Toc33018699)

[1.2.3 Roles and responsibilities 4](#_Toc33018700)

[1.2.4 Suspension Criteria and resumption requirement 4](#_Toc33018701)

[2. Testing strategy 4](#_Toc33018702)

[2.1 Functional testing 4](#_Toc33018703)

[2.2 User Interface testing 4](#_Toc33018704)

[2.3 Glass Box testing 4](#_Toc33018705)

[3. Resources and environment 5](#_Toc33018706)

[3.1 Testing tools 5](#_Toc33018707)

[3.2 Testing environment 5](#_Toc33018708)

[4. Approvals 5](#_Toc33018709)

# **Introduction**

This test plan is designed to describe what to be tested in the system, strategies to be used and the types of testing to be performed. This plan will identify features to be tested and tool that will be used to perform tests.

# Objectives

## Purpose

This document describes the plan for testing the AskAmech question and answer System.

This Test Plan document supports the following objectives:

* Identify test plan to be followed.
* To identify areas of the system to be tested.
* How the test should be performed and process to identify bugs and fix those bugs.
* The criteria to be used to identify if a test has failed instead of pass.
* Specific tasks to perform.
* List the recommended test requirements.
* To describe testing strategies to be used.
* Identify the required resources to perform tests.
* To identify critical performance measures.

## Scope and level of testing

Not all features of AskAMech which were defined in system specifications will be tested.

* Glass box testing

This testing will be done for each task that is implemented in the system.

Not all methods will be tested, but only those that produce an output to the user.

* Functional testing will be performed to check the functions of application. The functional testing will be carried out by providing the input and validates the output from the application.

This testing will be carried out after each implemented task to identify any bugs.

* User interface testing, to check if all system requirements were met we will navigate through the system to check if all functions and specification are implemented.

All functions of the system will be tested.

* The most critical performance measures to test are:

Request actions that are performed in the system (retrieving data from the database).

## Features to be tested

|  |  |
| --- | --- |
| **Feature name** | **Description** |
| Ask Question | User: a logged in user can ask a question in a website. |
| View Questions | All users: all users can view questions. |
| Answer to question | User: a user can provide an answer to a question. |
| Mark Answer as accepted Answer | User: An Author of a question can mark an answer as accepted. |
| Update Question | User: An Author of a question can update a question. |
| View User Profile | User: User can view profile for other users in the website. |

## Features not to be tested

These features are the system specifications document but they come with Entity Framework that were used to develop the system.

* Register
* Login

## Roles and responsibilities

All Team member will be responsible for createing test cases and test the system.

## Suspension Criteria and resumption requirement

If the team members report that there are **40%** of test cases **failed**, suspend testing until the development team fixes all the failed cases.

# Testing strategy

## Functional testing

## 

To ensure proper application navigation, data entry, processing, and retrieval, the following strategies will be used in functional testing together with SQL database:

* In functional testing the retrieval of data will show if the data retrieved is valid or not.
* Provide correct data entry and retrieve output to see if the correct data was processed by the system.
* Provide incorrect data to check if the validation is correct.
* The expected results occur when valid data is used.
* The appropriate message is displayed when invalid data is used.

## User Interface testing

To verify a user’s interaction with the system and ensure the system is working as intended, the following strategies will be used:

* html elements e.g. buttons, icons, searches and checkboxes will be clicked and verify if they are invoking correct actions in the system.
* The system will be tested in different browsers to conform if all elements are visible and if they are visible in a desired position.

## Glass Box testing

They are errors that can be eliminated before trying to perform a functional testing, the following strategies will be used:

* Test-Driven-Development (TDD) will be used to test methods of the system if they are behaving in a correct way.

# Resources and environment

## Testing tools

NUnit

NUnit3TestAdapter

NSubstitute

PeanutButter.RandomGenerators

## Testing environment

A windows environment with internet explorer 8,9 and 10 and Google Chrome.

# Approvals

|  |  |
| --- | --- |
| **Name** |  |
| **Signature** |  |
| **Date** |  |