Software Requirements Specification

1. Introduction

1.1 Purpose

The purpose of this document is to build an online test system to check and improve road rules knowledge by passing tests. System will allow user to register and control their progress by checking statistics, based on number of passed and failed tests.

1.2 Document conventions

ASP	Active Server Pages
DB	Database
API	Application Programming Interface
ER	Entity Relationship
UML	Unified Modelling Language

1.3 Intended audience and reading suggestions

This document describes a system for road rules knowledge testing. It will be useful not only for people who want to get the driver license, but also for experienced drivers. Overall, the system will add more safety to the traffic, as people tend to forget what they learnt a long time ago, so refreshing this knowledge will help them to make an important decision in a nonstandard situation. The system stores statistics of passed and failed tests, so it may be applicable for driving schools also.

1.4 Project scope

The system is a web-based 3-tier application, based on 2 databases:

- Relational database to store tests-related data (answers, options, pictures for tests) and statistics-related data;
- Relational database to store user-related data (users, their roles, profile pictures etc).

The purpose of the system is high convenience and reliability. It will be easy-to-use for the majority of social groups that need road rules knowledge development.

System will support flexible test sets of different complexity to provide comfortable user experience for administrators and end users.

2. Overall Description

2.1 Product Perspective

The road rules testing service offers such features:

- **Database of set tests and questions, which can be updated by administrator.** - This feature allows administrators to update existing tests when, for example, government introduces some changes or updates to existing road rules.
- **Statistics of number of passed and failed tests available for the end user.** - Registered users can track their progress by checking statistics of successfully passed and failed tests for each area.
- **Questions categorizing - user can see a label of question for its better understanding.** - User can mark failed or complicated for them questions to revise a particular section of road rules. This will make learning less complex, as user will always see the weak areas of their knowledge.

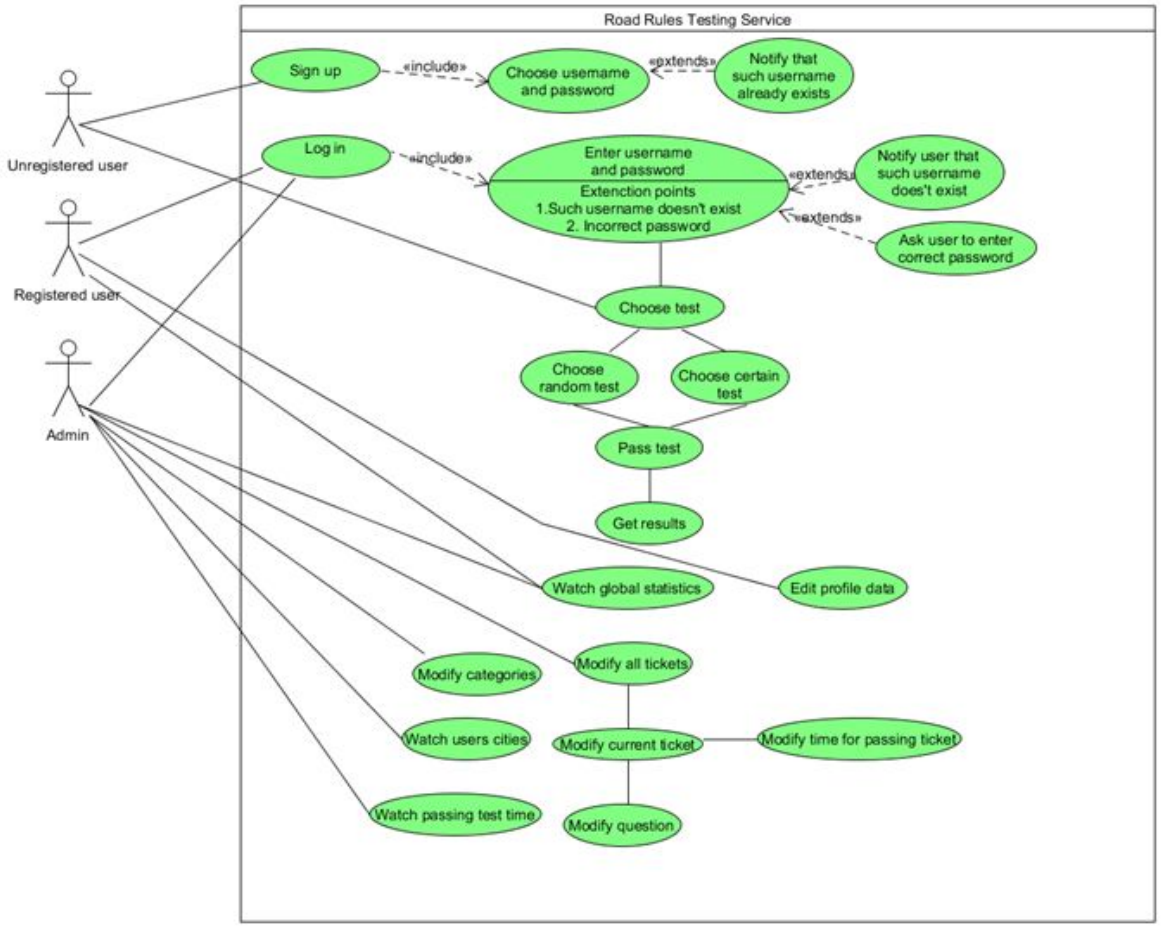
2.2 Product features

- Unregistered user can sign up in the system. During this process user have to choose username and password.
    - When user chooses an existing username he will be notified about it and will be asked to choose another login.
    - User enters new username and chooses password he is successfully registered in system.
- Registered user can log in. To do this, user have to enter username and password.
    - In case of entering wrong username or password user will be notified about it and will be asked to try to enter correct username/password.
    - In case of entering correct username and password user is logged in.
- Either registered or unregistered users can pass test. Test can be chosen
    - randomly;
    - from given list.
- Each ticket has set time to pass it.
    - User can end testing before time is out.
    - If time is out test is automatically ended.
- After finishing test user can watch results.
- Registered user can edit his profile.
- Registered user can watch his statistics, that includes total percentage of correct and wrong answers.
- Admin can log in, pass test and watch statistics like any other registered user.
- Admin can watch users cities and time when users passed tests.
- Admin can modify categories of questions.
- Admin can modify questions and tickets.
    - Modifying ticket includes replacing a question by another from the existing list.
    - Question can be modified in two directions

- modification the text of question
- modification of answers.

2.3 User class and characteristics

User classes and characteristics can be demonstrated by use case UML diagram:



Use cases description:

№	Name	Functionality	Description
1	Sign up	Creates new account	User creates new personal account and fill it with relevant information.

2	Log in	Sign in to existing account	User takes to his personal account via login and password.
3	Choose test	User chooses one of possible test	User can start random test or choose certain ticket.
4	Passing test	User is passing test	User is asked to give answer to different questions. After answering question user sees the correctness of the answer.
5	Get result	User gets current test result	After passing test user can see percentage of correct, wrong, and not answered questions.
6	Modify question	Admin change the content of question	Admin can change question text, picture, complexity, category of question and set of possible answers.
7	Modify ticket	Admin change ticket content	Admin can change ticket's question set, allowed mistakes percent and time period for each ticket passing.
8	Modify category	Admin can modify questions categories	Admin can add and delete question categories
9	Modify profile	Modify profile data	Any registered user can modify his profile data
10	Modify all tickets	Add and delete tickets	Admin can add and delete tickets
11	Watch global statistics	Watch all time statistics	Any registered user can watch his overall statistics, considering all previous testing

12	Watch users data	Watch cities and time of passing test	Admin can watch cities users are from and time when they passed test

2.4 Operating environment

Operating environment for road rules testing system is listed below:

- Database: MS SQL Server databases for user identity and tests storage.
- Client/server: ASP.NET MVC / .NET Core
- CI/CD Environment: Travis CI

2.5 Assumptions and dependencies

The main assumption that is made before this project start is the willingness of people to learn road rules, as in some regions people may get the driver license without any knowledge of road rules, so this service will not be popular in such countries. Also there are still some people that are not very familiar with modern technologies, so tests on the paper may look more convenient to them that the online ones.

The service will be self-contained without significant dependencies on third-party vendors. The system will require a web-hosting environment to be always online and accessible for the end users.

3. System Features

3.1 Description and priority

The main idea of this project is the development of a convenient and simple system for people to evaluate and improve their road rules knowledge. The system may be used by everyone who has or wants to have a driver's license, so end-user categories are very different. The main priorities of the project are simplicity, convenience, and intuitiveness of the UX part. It will make the system easy to use for everyone.

3.2 Stimulus/response sequences

User creates new personal account and fill it with relevant information. User takes to his personal account via login and password. User can start random test or choose certain ticket.

User is asked to give answer to different questions. After answering question user sees the correctness of the answer.

Admin can change question text, picture, complexity, category of question and set of possible answers. Admin can add and delete question categories. Admin can add and delete tickets. Admin can watch cities users are from and time when they passed test

- **Stimulus:** Sign up with new login and correct password
- Response: The user created
- **Stimulus:** Sign up with taken login
- Response: The new user is not created, error displayed.
- **Stimulus:** Select answer
- Response: Selected answer highlighted
- **Stimulus:** Sign out

- Response: The session for current user is closed
- **Stimulus:** Confirm wrong answer
- Response: Correct answer hightlighet with green, wrong answer hightlighet with red

3.3 Client/Server system

The term client/server refers primarily to architecture or logical division of responsibilities, the client is the application (also known as the front-end), and the server is the DBMS (also known as the back-end).

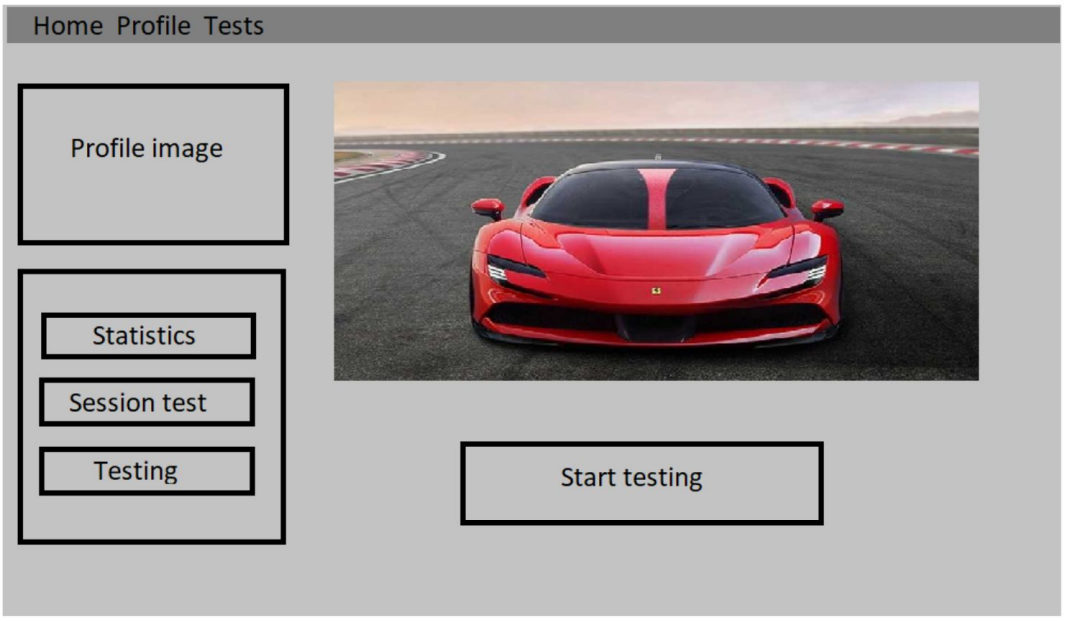
A client/server system is a distributed system in which,

- Some sites are client sites and others are server sites.
- All the data resides at the server sites.
- All applications execute at the client sites.

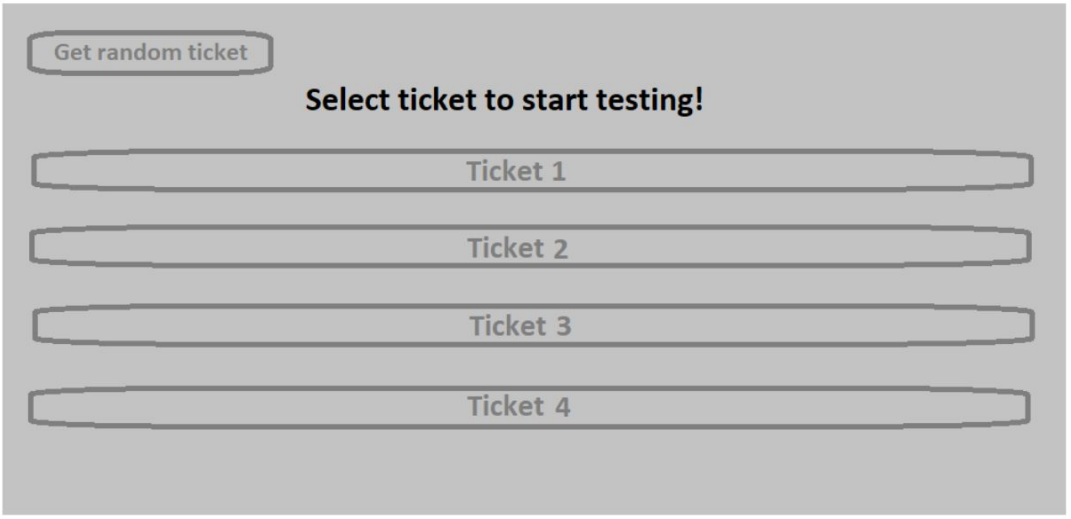
4. External Interface Requirements

4.1 User Interfaces

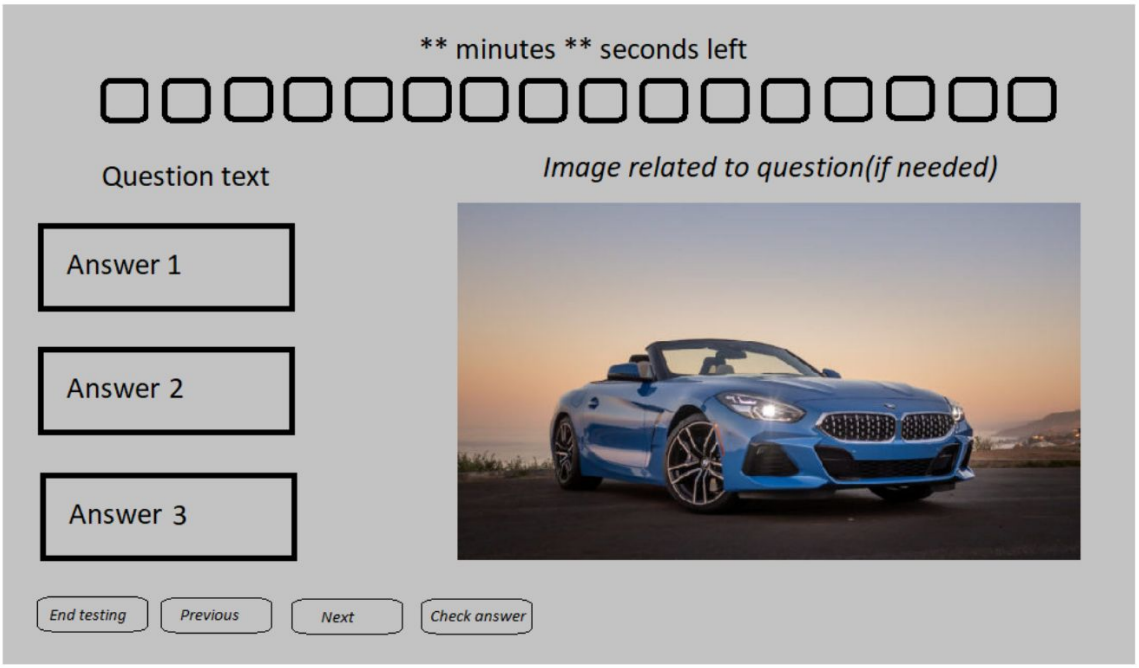
The following mock-ups describe most features of user interface.
Homepage



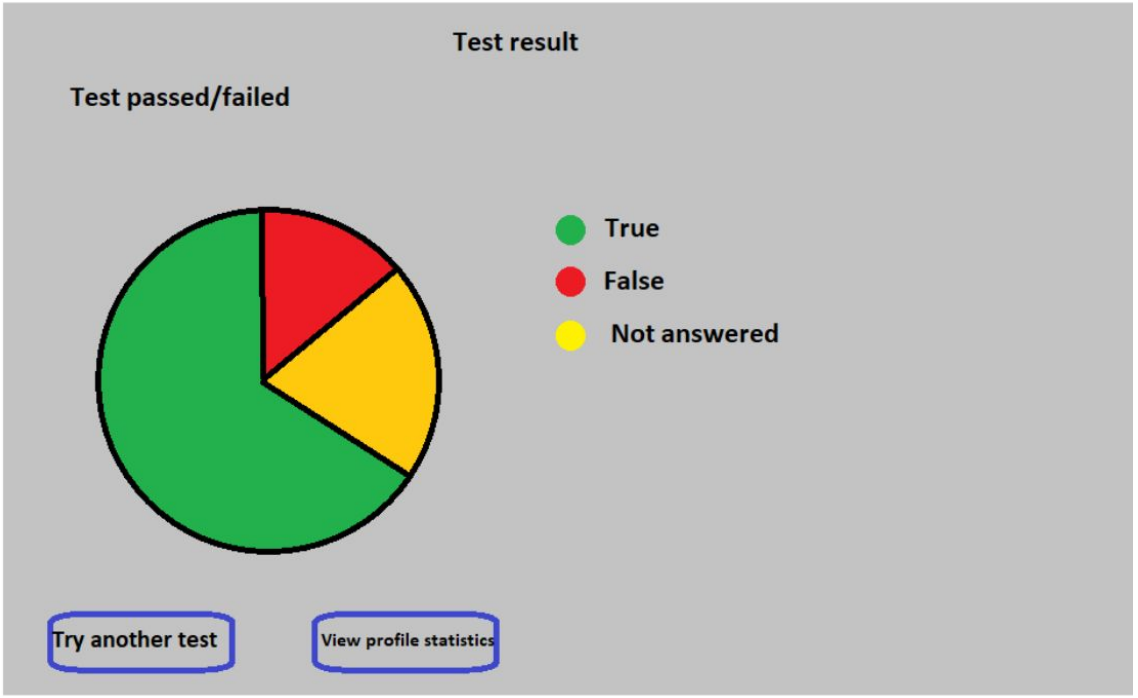
Select ticket view



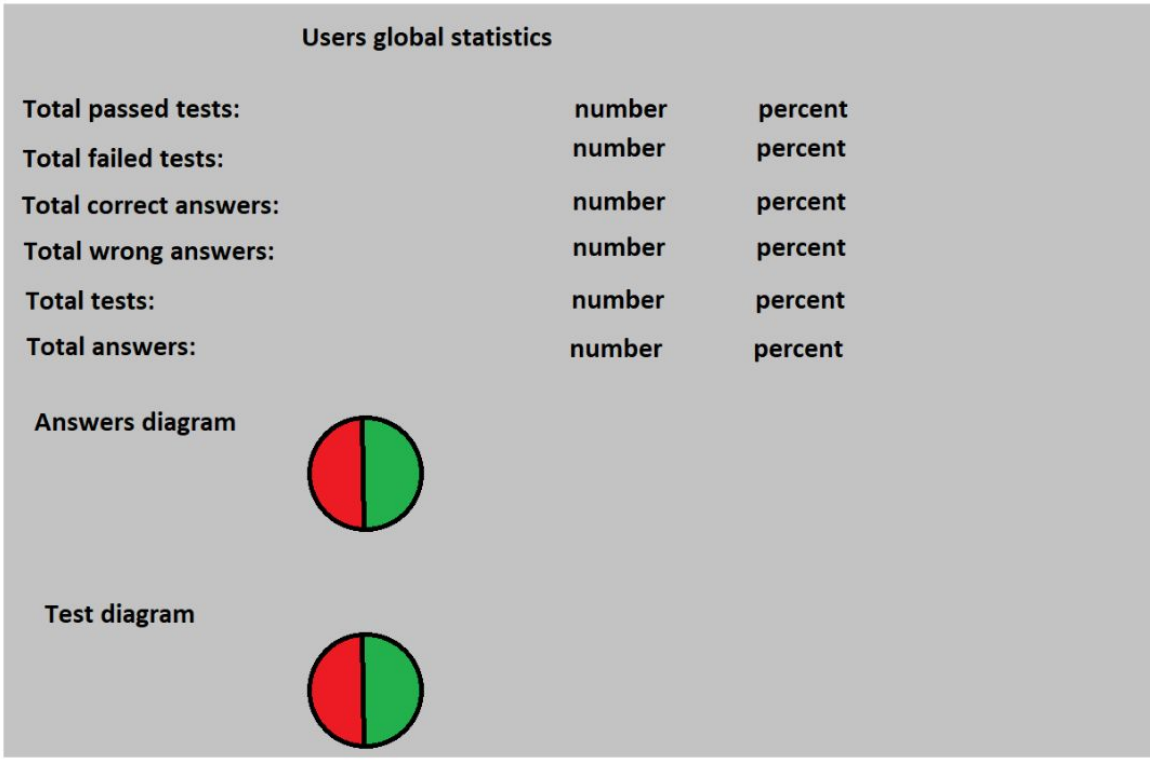
Question in test page



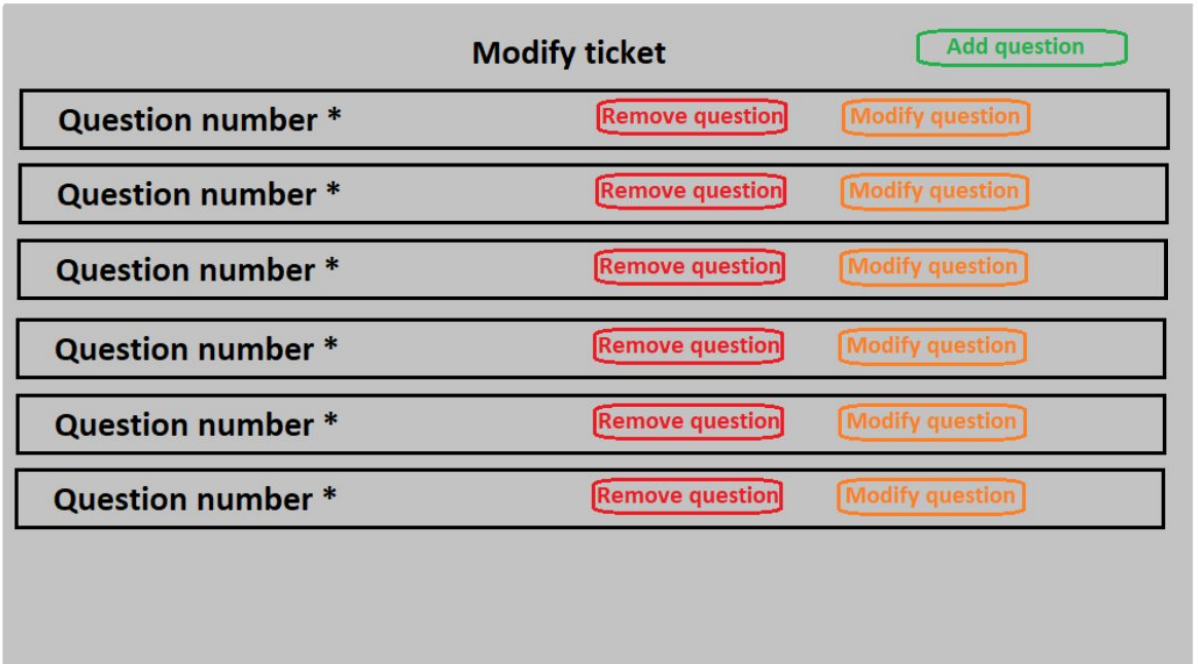
Test results



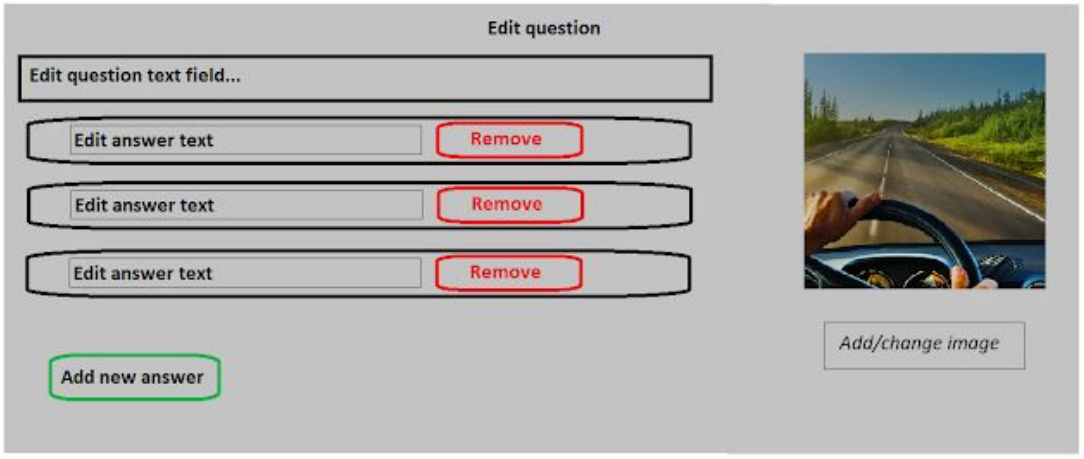
User global statistics view



Ticket modification view



Question editing



4.2 Hardware Interfaces

Computer with Windows

4.3 Software Interfaces

Software used	Description
Operating system	Windows, chosen for its best support and user-friendliness
Framework	.Net framework, most suitable framework for current task
Database	MS SQL, can be easily integrated with .Net

4.4 Communication Interfaces

This project will support all types of modern web browsers(Google Chrome, Mozilla, Edge, Internet explorer).

5. Nonfunctional Requirements

5.1 Performance Requirements

This application is an online test system to check and improve road rules knowledge by passing tests. As we know there is some amount of time to pass the whole test. It is very important to avoid delay while loading the next question or processing the user answer (especially when there are a lot of users passing tests simultaneously.

5.2 Safety Requirement

The test system can be used in several scenarios: for education purposes and for government exam purposes. There are several safety issues in both cases. If administration stuff will not care about application content it may lead to the wrong knowledge of the new drivers. It can be a crucial factor in the case of government exams. Wrong knowledge of the road rules can lead to catastrophic failure.

5.3 Security Requirement

The test system potentially can be used by the government in order to test new drivers before the issuance of driver's licenses. In that case, security is a crucial factor in several cases:

- Exclude possibility to change an answer after confirmation
- Exclude possibility to change test final score
- Exclude possibility to access database (with all question and answers)
- Exclude the possibility to modify testing results by administration stuff (e.g. corruption case)

It can be done by using modern security technology, cryptography and proper support of this application.

5.4 Software Quality Attributes

- Availability:` The testing system should be available on the specified date and specified time. Otherwise, there will be serious inconveniences up to exam cancelation.
- Correctness: The testing system should contain only valid questions and answers.
- Maintainability: The administrators should maintain tests database, categories, questions, tickets, etc. But no one could be able to edit testing results, prepare specific questions, tickets for certain users.
- Usability: The testing system should satisfy the maximum number of customers' needs.

5.5 Acceptance criteria

Scenario: Sign up with new login and correct password

Given	The login has not been taken and password correct
When	User is not signed in
Then	The new user created

Scenario: Sign up with new login and invalid password

Given	The login has not been taken and password invalid
When	User is not signed in
Then	The new user is not created, error displayed

Scenario: Sign up with taken login

Given	The login has not been taken
When	User is not signed in
Then	The new user is not created, error displayed

Scenario: Sign out

Given	Sign out button pressed
When	User is signed in
Then	The session for current user is closed

Scenario: Chose ticket

Given	The user is optionally signed in
When	Ticket page displayed
Then	Page with ticket description displayed

Scenario: Start ticket

Given	The user is optionally signed in
When	Ticket is selected
Then	First question of the ticket displayed, timer started

Scenario: Select answer

Given	Ticket is going
When	User select one of the given answers
Then	Selected answer highlighted

Scenario: Confirm correct answer

Given	Ticket is going, answer is selected
When	User confirm correct answer
Then	Answer hightlighet with green

Scenario: Confirm wrong answer

Given	Ticket is going, answer is selected
When	User confirm wrong answer

Then	Correct answer hightlighet with green, wrong answer hightlighet with red

Scenario: Skip answer

Given	Ticket is going
When	User skips question
Then	The next question is displayed

Scenario: Signed out user finish ticket

Given	Ticket is going
When	User press finish button
Then	Ticket statistics are displayed (correct, wrong, not answered). Final result displayed

Scenario: Signed in user finish ticket

Given	Ticket is going
When	User press finish button
Then	Ticket statistics are displayed (correct, wrong, not answered). Final result displayed, user profile statistics updated

Scenario: View statistic

Given	User is signed in
When	User open his profile
Then	User overall statistic displayed

Scenario: Admin signed in

Given	User sign in with admin account
When	User is signed out
Then	Home page displayed with admin tool

Scenario: Open admin tool

Given	User signed in as admin
When	Admin tool opened
Then	Admin home page displayed

Scenario: Admin add new ticket

Given	Admin page opened
When	New ticket button pressed

Then	New ticket page opened

Scenario: Admin add question

Given	Admin open one of the tickets
When	New question button pressed
Then	New question is created, admin asked to fill up question details