



«Московский государственный технический  
университет  
имени Н.Э. Баумана»  
(МГТУ им. Н.Э. Баумана)

# Курсовой проект по курсу «Базы данных» на тему: «Система тестирования»

Студент:

Анисимов Никита Сергеевич

Руководитель:

Строганов Юрий Владимирович

# Цель и задачи

## Цель

- Разработать систему автоматического тестирования.

## Задачи

- Проанализировать существующие решения;
- Выбрать подходящий инструментарий;
- Разработать приложение.

# Существующие решения

- Яндекс.Контест;
- Сертификация Mail.ru;
- Stepik;
- Онлайн олимпиады.

# Требования

- Регистрация пользователей;
- Изменение данных аккаунта и личных данных;
- Создание новых тестов;
- Удаление созданных тестов;
- Поиск тестов;
- Просмотр ранее пройденных тестов;
- Прохождение тестов.

# Базы данных

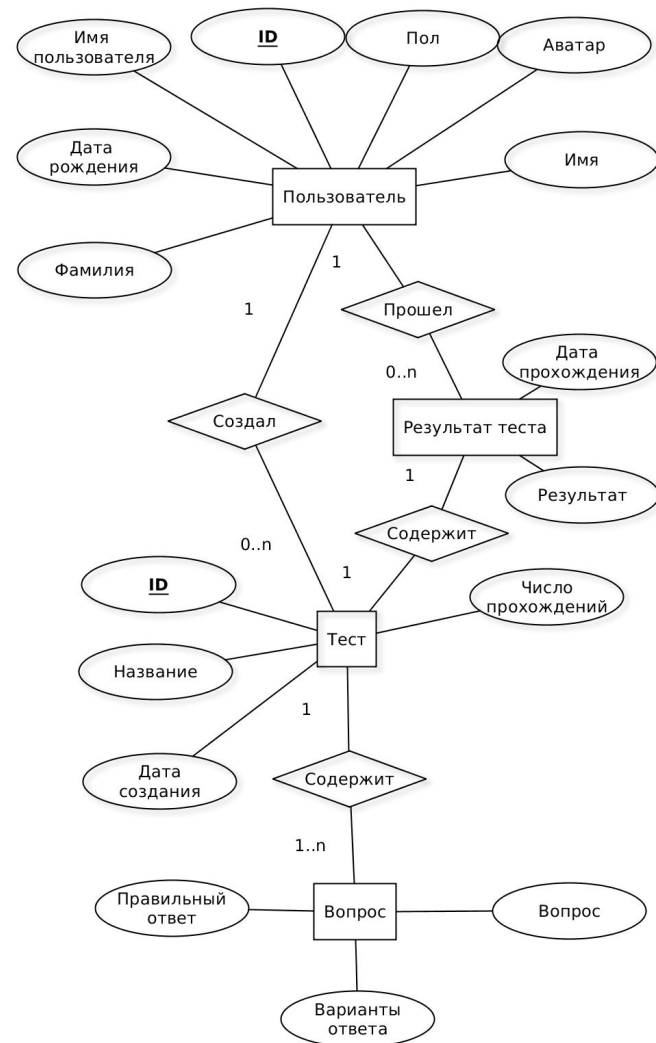
SQL	NoSQL
<b>Структура и тип хранящихся данных</b>	
Требуется наличие однозначно определенной структуры хранения данных	Нет ограничений на структуру данных
<b>Запросы</b>	
Язык SQL	Каждая NoSQL база данных реализует свой способ работы с данными.
<b>Масштабируемость</b>	
Вертикальное масштабирование выполняется за счет увеличения числа системных ресурсов. NoSQL обычно предоставляют более простые способы горизонтального масштабирования.	

# Диаграмма прецедентов



# Сущности

- Пользователь
- Тест
- Результат теста
- Вопрос



# Разработка

- Сервер
  - Haskell
  - Servant
  - Persistent
- Клиент
  - Elm



# Haskell

- Автоматическое управление памятью;
- Чистые функции;
- Ленивая модель вычислений;
- Параметрический полиморфизм;
- Параллельные вычисления.

# Servant

## API

```
type QuizAPI = "new"  
  :> ReqBody '[JSON] Quiz  
  :> Post '[JSON] Result
```

## Обработчик

```
newQuiz :: ServerT QuizAPI  
newQuiz quiz = ...
```

# Persistent

## Описание модели

```
mkPersist sqlSettings [persist|  
Person  
    name String  
    age Int  
    deriving Show  
|]
```

## Использование

```
func ... = do  
    ...  
    insert $ Person "Michael" 26  
    ...
```

# Elm

- Модель
- Обновление
- Представление

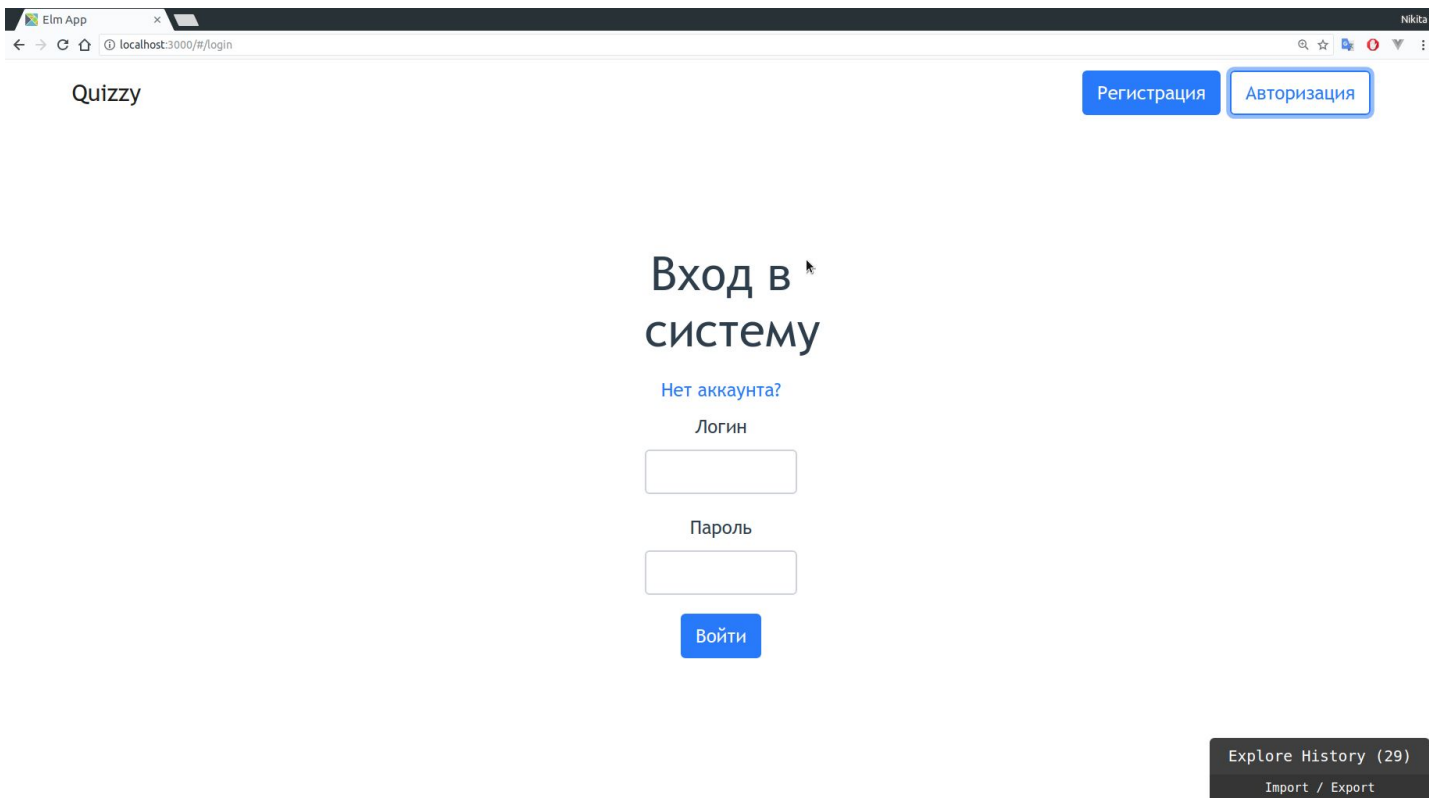
```
type alias Model = { ... }
```

```
type Msg = Reset | ...
```

```
update : Msg -> Model -> Model  
update msg model =  
case msg of  
Reset -> ...
```

```
view : Model -> Html Msg  
view model = ...
```

# Интерфейс авторизации



The screenshot shows a web browser window with the title 'Elm App' and the URL 'localhost:3000/#/login'. The browser's address bar shows the URL. The page has a dark header bar with the name 'Quizzy' on the left and two blue buttons, 'Регистрация' and 'Авторизация', on the right. The 'Авторизация' button is highlighted with a blue border. Below the header, the main content area is white. It features the text 'Вход в систему' in a large, bold font. Below this text is a link 'Нет аккаунта?' in blue. Underneath the link are two input fields: the first is labeled 'Логин' and the second is labeled 'Пароль'. Below the input fields is a blue button labeled 'Войти'. In the bottom right corner of the page, there is a dark gray button labeled 'Explore History (29)' with the text 'Import / Export' below it.

Quizzy

Регистрация Авторизация

Вход в систему

[Нет аккаунта?](#)

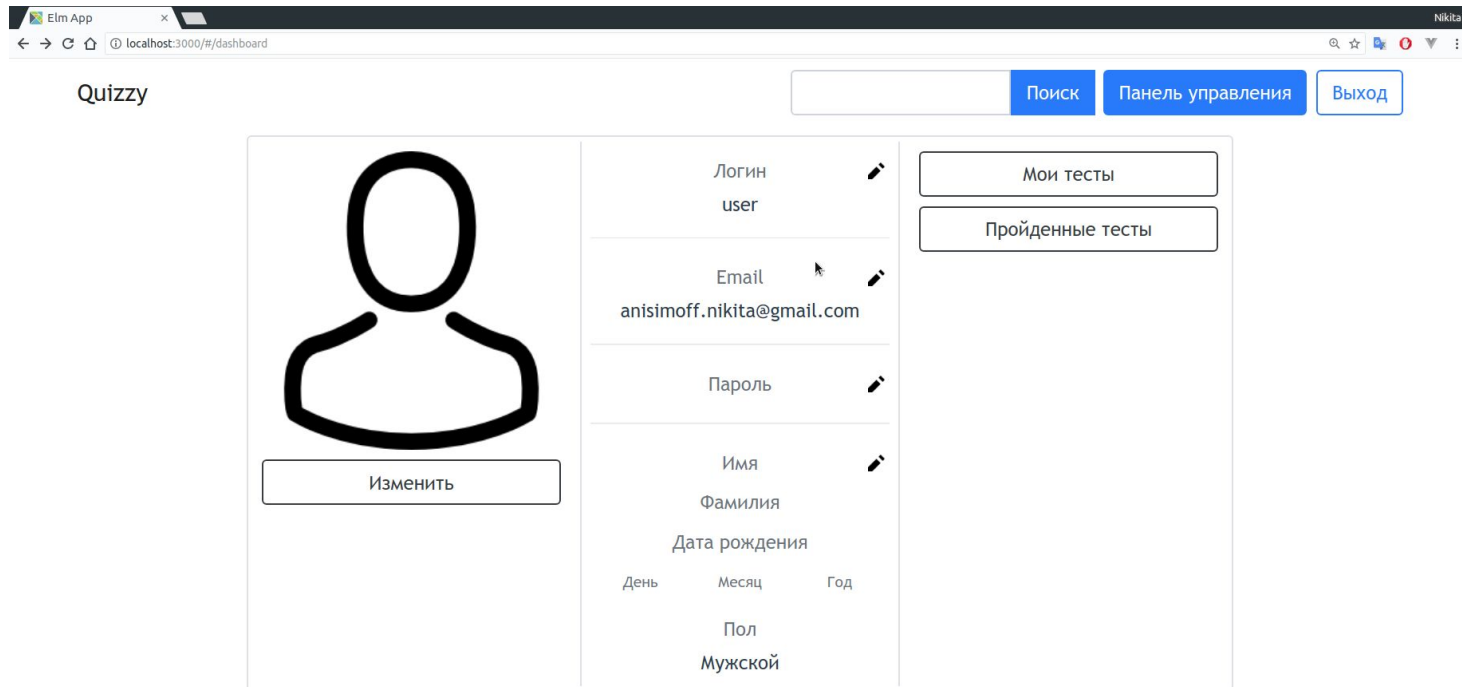
Логин

Пароль

Войти

Explore History (29)  
Import / Export

# Интерфейс страницы пользователя



# Вывод

В результате выполнения работы был:

- проведен анализ существующих решений;
- подобран подходящий инструментарий:
  - Сервер: MongoDB, Haskell, Servant, Persistent
  - Клиент: Elm
- разработан сервер;
- разработано клиентское приложение.