Міністерство освіти і науки України

Українська академія друкарства

Кафедра АКТ

**Звіт**

з виробничої практики

на підриємстві

ТОВ “Exoft”(Ексофт)

Виконав:

Студент групи АУ-4

Майхер Мар’ян

Прийняв:

ст.викладач

Шевчук О.В.

Львів 2016

Зміст

* Вступ
* Скаладання вимог до проекту
* Розробка архітектури для заданого рішення
* Технічна реалізація рішення і супутнє тестування
* Завершальний етап, хостинг.
* Висновки

***Вступ***

Підприємство ТОВ “Ексофт” активно займається розробкою програмного забезпечення (аутсорсингом) для найрізноманітніших рішень і галузей промисловості. У своїй роботі працівники компанії переважно користуються стеком технологій від Microsoft, а саме

* .Net framework
* WPF(Windows Presentation Foundation)
* WinRT(Windows runtime)
* ASP MVC(Active Service Pages Model View Controller)
* ASP Web Forms
* JavaScript(AngularJs, Jquery, ReactJs, D3Js, KnockoutJs)
* HTML5, CSS
* Xamarin

Офіційний сайт підприємства можна відвідати за адресою <http://exoft.net/>.

Компанія активно залучає до розробки молодих амбіційних розробників, а також впроваджує застосування нових технологій які ніколи не стоять на місці.

***Складання вимог до проекту***

Проект носить назву eQuiz. В сутності це система присзначена для автоматизації процесу проходження віддаленого тестування кандидатів. Для забезпечення налагодження правильного циклу роботи системи передбаччено три ролі користувачів, а саме:

* Модератор(в його функції входить створення нових, видалення та зміна існуючих тестів)
* Адміністратор(перевіряє правильність відповіді на неавтоматичні питання, виставляє оцінку користувачу, зберігає результат в базі даних)
* Студент(аутентифікуєься в системі, проходить тест, відповідає на питання, зберігає свій результат)

Відповіно до трьох основних частин проекту команда розробників ділиться на три частини. Частини відносно незалежні одна від одної, тому процес розробки вимагає тільки мінімальної комунікації між командами.

Я був залучений до роботи з командою яка відповідала за студентську частину, тобі надалі в звіті детально буде описуватися частина студентів.

Детальні вимоги до студентської частини:

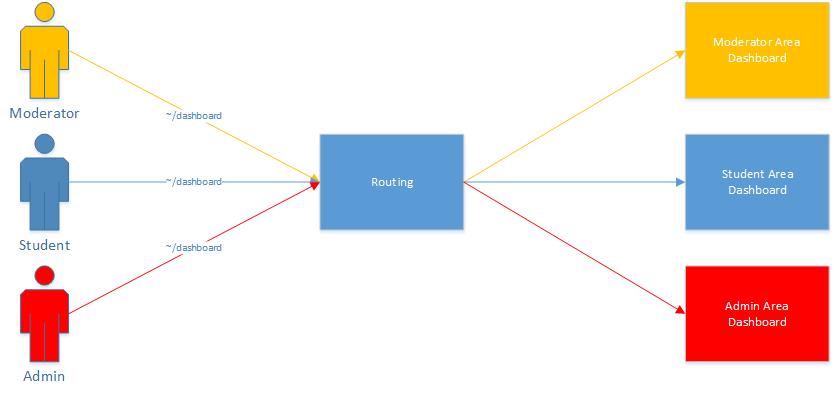
**Підхід до рішення**

Система повинна бути реалізована у вигляді монолітного веб-додатку на основі стеку технологій ASP.NET. Він повинен складатися з 2-х шарів: презентаційного рівня і доступу до даних. Кожен шар повинен бути реалізований у вигляді окремого проекту.

**Рівень представлення**

Презентацій шар повинен бути побудований на ASP.NET MVC 5. Angularjs (з ASP.NET MVC JSON або WebAPI)

ASP.NET MVC Області (поряд з маршрутизацією на основі обмежень) повинні бути використані для реалізації різних, специфічних для ролей, пердставлень:



Всі загальні представлення повинні бути винесені з областей.

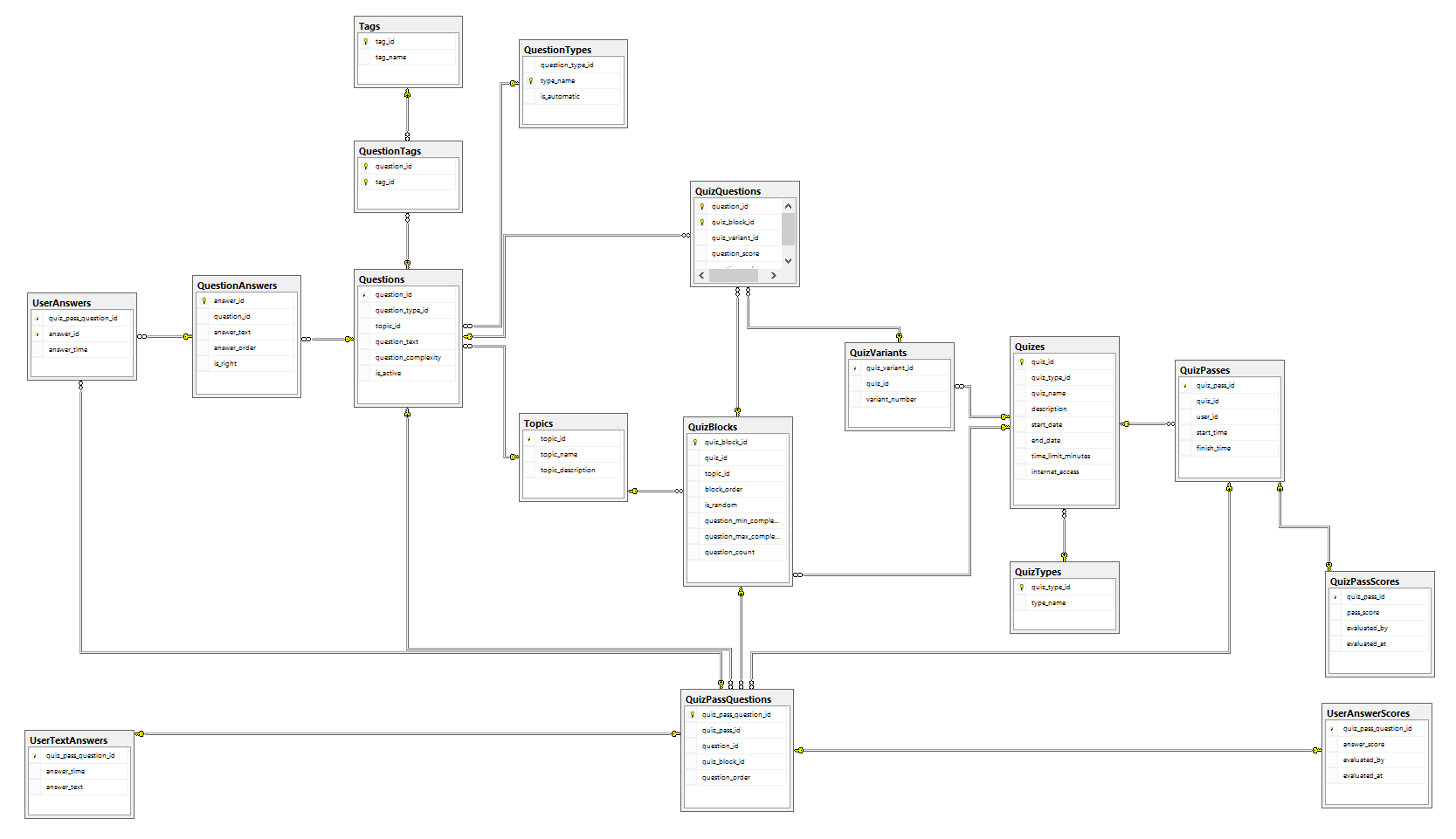
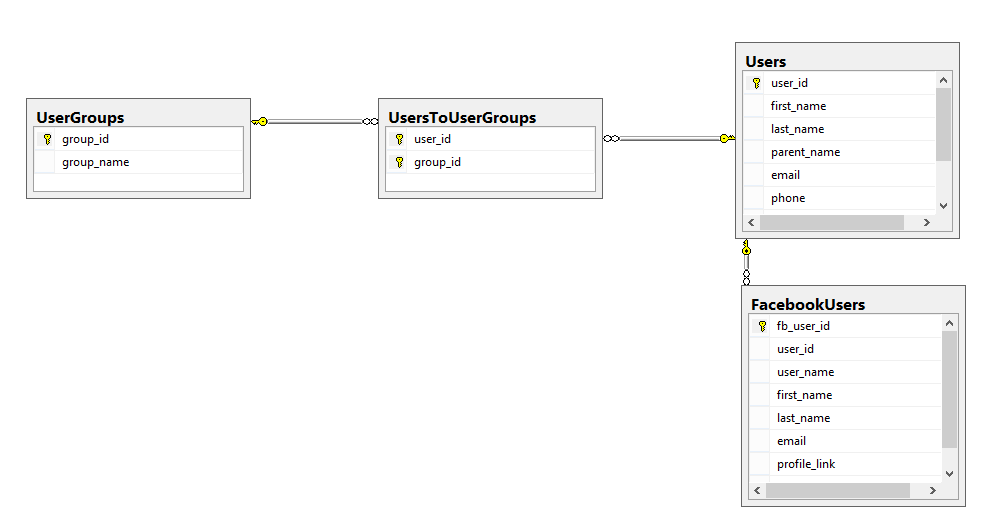
Всі сторінки за замовчуванням повинні мати фільтрацію. AJAX повинен використовуватися для всіх запитів до даних сторінки (фільтрація, сортування,та груповання).

**Рівень доступу до даних**

Рівень доступу до даних повинен реалізуватися за допомогою репозиторіїв. Всі взаємодії з базою даних повинні бути реалізовані за допомогою технології Entity framework.

Схема бази даних вже розроблена. За нею потрібно слідувати в процесі розробки. Вона може бути зміненена тільки в разі потреби. Всі такі зміни повинні бути розглянуті і затверджені.

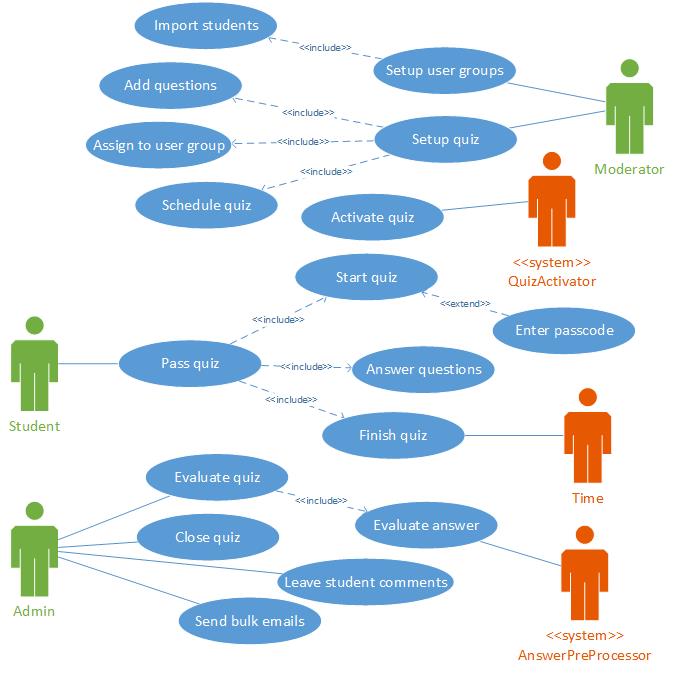
Нижче приведена схема бази даних:



**Архітектура рішення**

Цей розділ охоплює архітектуру.

Сценарії:



Система підтримує три типи користувачів:

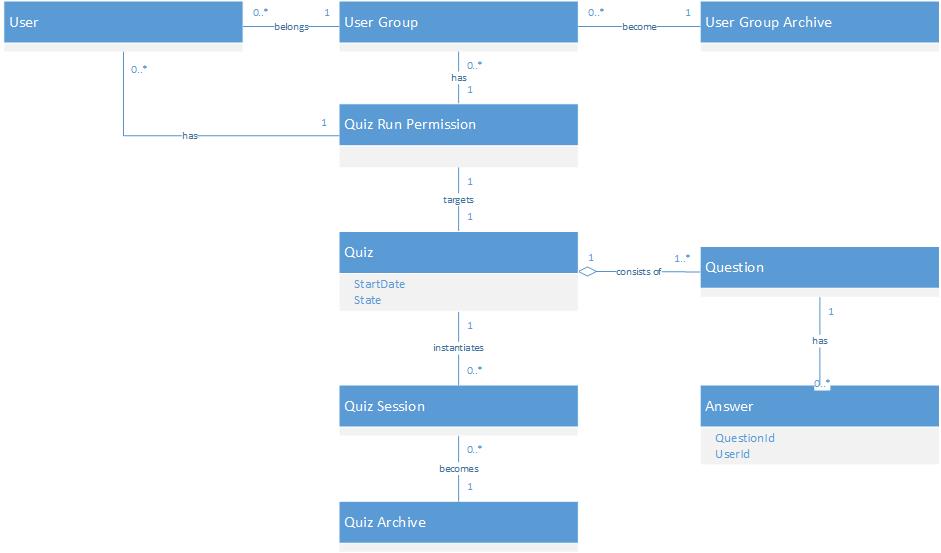
• Модератор - відповідальний за управління групами користувачів, створення тесту та планування.

• Студент - може пройти тест, відповідаючи на питання.

• Адміністратор - відповідає за оцінювання та зберігання тесту.

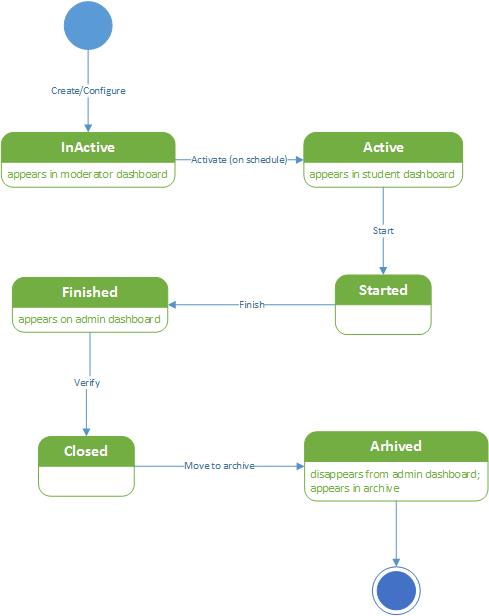
**Доменна модель**

Модель предметної області показана нижче



**Життєвий цикл тесту:**

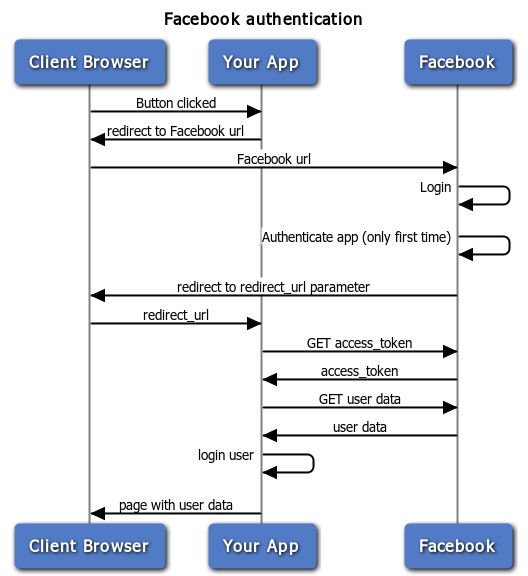
Тест є серцем системи і всі функціональні можливості мають бути побудовані навколо нього. Тест повинен бути підготовлений модератором, щоб стати видимим для студента. Студент намагається пройти тест, відповідаючи на питання. І, нарешті, студентські відповіді оцінюються адміністратором. На наступному малюнку показані основні етапи тесту:



**Безпека**

Система повинна підтримувати SSL (HTTPS). Всі запити HTTP повинні бути перенаправлені на HTTPS. Тільки зареєстровані користувачі матимуть доступ до системи. 3 ролі повинні бути підтримані: модератор, студент і адміністратор.

Аутентифікація повинна здійснюватися на основі Owin. Система повинна забезпечувати 2 окремі режими: один для студентів, інша - для модераторів і адміністраторів. За замовчуванням логін сторінка повинна дозволяти входити тільки в якості студента. Тільки облікові записи соціальних мереж можуть бути використані для перевірки автентичності. Facebook і Google, а також LinkedIn провайдери аутентифікації повинні бути підтримані. OAuth 2.0 слід використовувати для інтеграції. Для прикладу показаний підхід аутентифікації facebook:



Основні фази роботи

Фаза 1 Область застосування і призначення:

• Команда 1 - функціональність студента

Вибір тесту, навігація по питаннях, відповіді на питання, завершення тесту, впровадження таймеру.

• Команда 2 - функціональність модератора

Управління тестами, створення, зміна, видалення.

• Команда 3 - функціональність адміністратора

Оцінювання тесту та його закриття

Інші завдання (загальні функціональні можливості):

• Налаштування EF Міграції

• Маршрутизація областей на основі установки ролі

Фаза 2

Фаза 2 Область застосування:

• студент вхід: використання facebook провайдера аутентифікації

студент вхід: використання LinkedIn провайдера аутентифікації

• студент вхід: використання провайдера аутентифікації Google

• студент: оновлення даних облікового запису

• модератор / адміністратор, вхід в систему: Exoft SSO

• модератор: управління групами користувачів

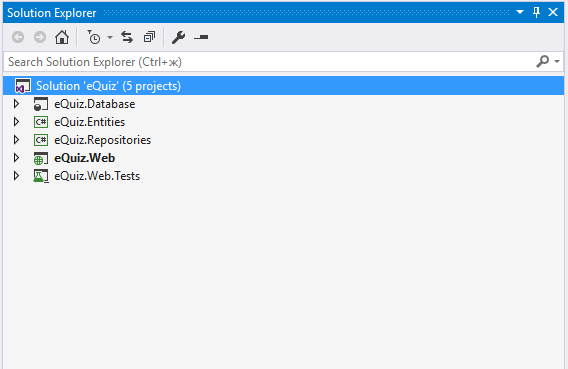
• адміністратора: додавання коментарів до пройденого тесту

• адміністратора: архівування результатів

***Розробка архітектури для задаого рішення***

Проект складається з декількох частин.

Архітектура приведена нижче:



Database – база даних, та скрипти для її створення, видалення та наповнення тестовими даними.

Entities – сутності з бази даних, приведені до об’єктно-орієнтованих класів, для зручності роботи.

Repositories – предствляє собою доступ до даних, тобто набір методів бля заємодії з об’єктами бази даних(CRUD-операції).

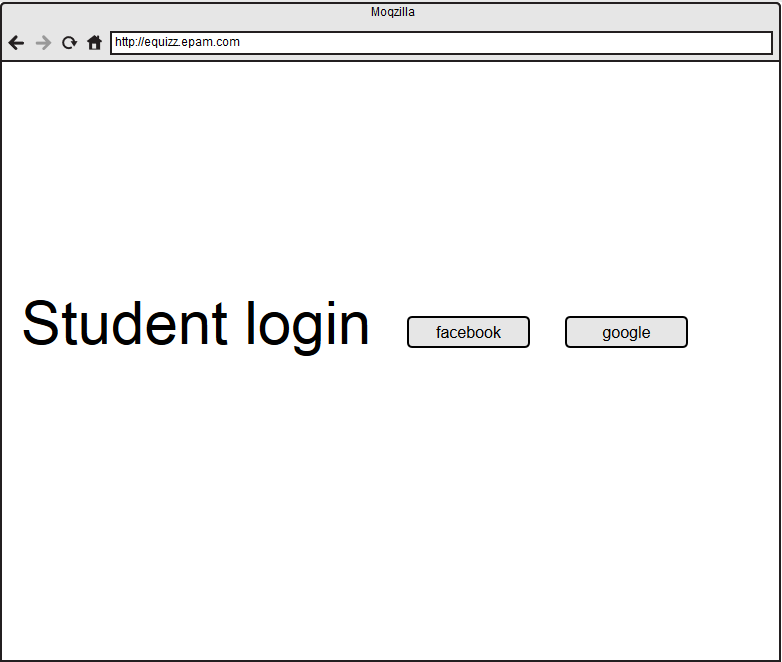
Web – головний проект, який являє собою власне веб-аплікацію, яка використовує інші проекти в роботі, містить графічний інтерфейс і всю основну бізнес-логіку програми.

Tests – проект з модульними тестами для полегшення командної розробки

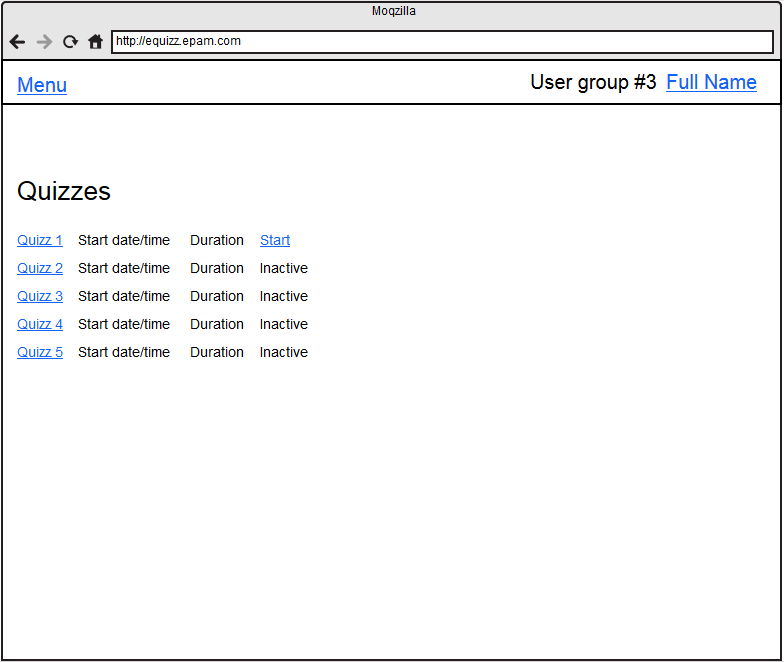
***Технічна реалізація рішення і супутнє тестування***

Попередні макети графічного інтерфейсу:

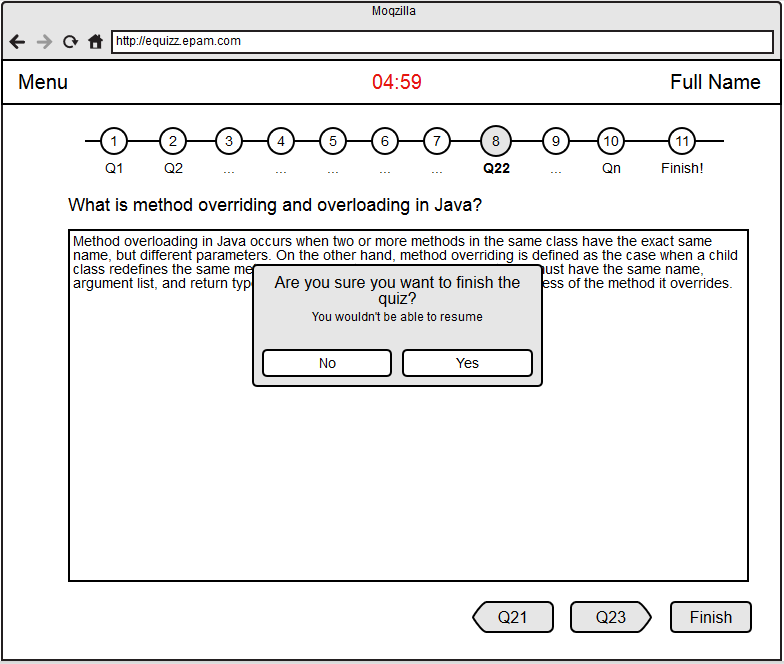
Сторінка аутентифікації



Сторінка вибору тесту:

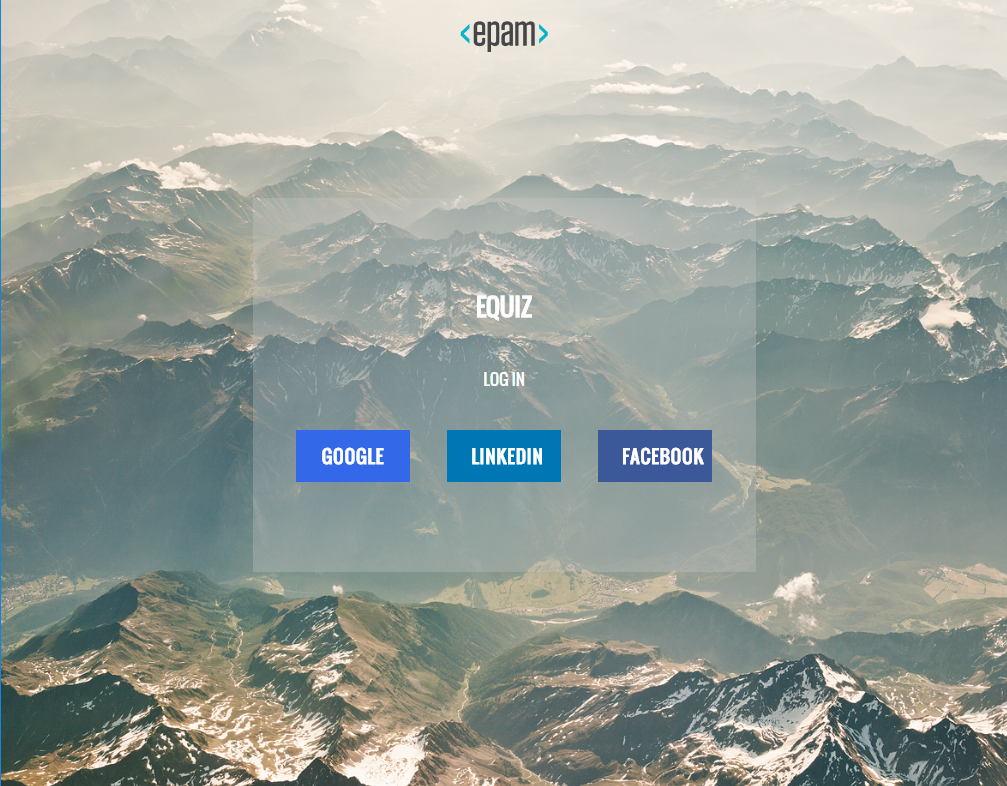


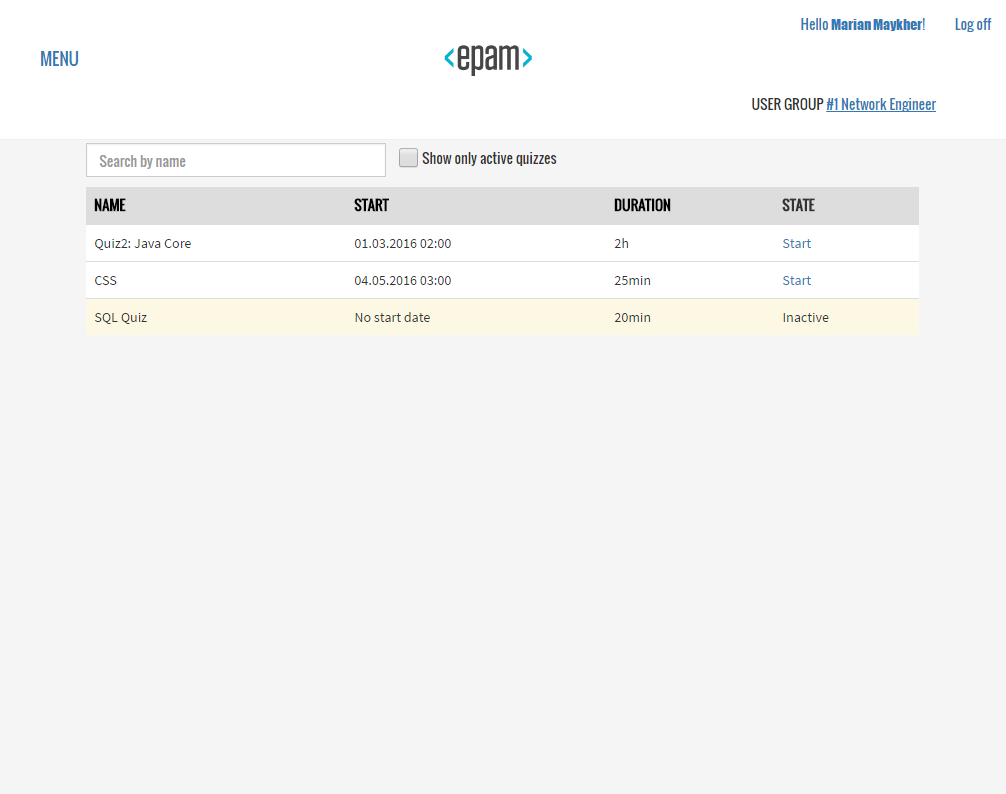
Сторінка проходження тесту:

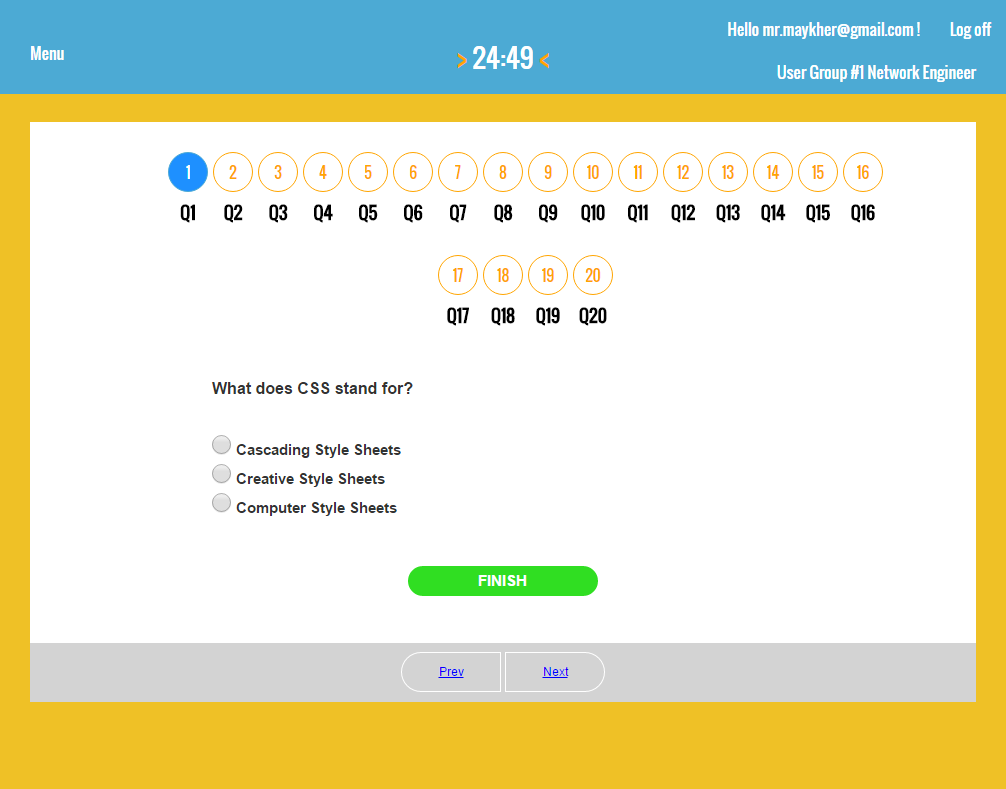


Головна ціль тестових макетів просто показати шаблон сторінки не проводячи стильове оформлення сторінок.

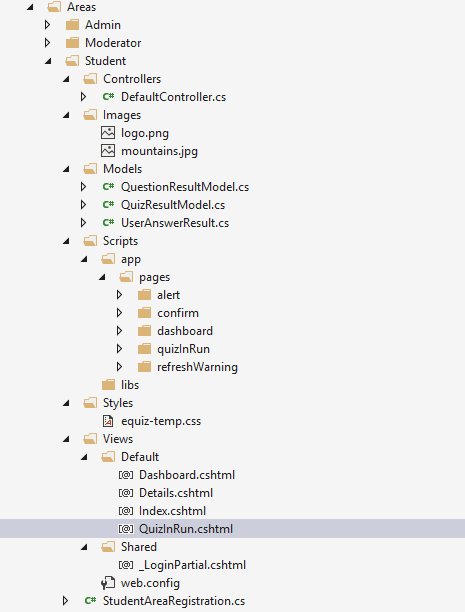
Нижче показано ті ж сторінки але вже з накладеним дизайном:







**Детальна архітектура студентської частини**:



На скрін-шоті приведено основні функціональні елементи студентської частини програми, а саме: у папці Controllers присутній головний контроллер DefaultController, який є серцем програми. Контроллер приймає запити від користувача і відповідним чином обробляючи їх видає потрібну відповідь.

У папці Images присутні дві фотографії які використовуються при побудові графічного інтерфейсу.

Папка Models містить набір класів для коректної обробки результатів які приходять від користувача.

Одна із ключових папок – Scripts містить реалізацю всієї клієнтської логіки проекту.

Папка Styles – відповідає за стильове оформлення графічного інтерфейсу.

Папка Views представля власне сам графічний інтерфес, набір сторінок, в який розміщаються основні елементи взаємодії з користувачем.

Даний набір компонентів є оптимальним для даного рішення, оскільки забезпечує реалізацію всіх вимог проекту а також компактність і структурованість вихідних кодів.

В процесі розробки критично важливо після проведених змін активно застосовувати ручне тестування функціоналу, на що в відповідному проекті виділялася майже третина всього часу. На щастя середовище розробки Visual Studio 2015 надає всі необхідні інструменти для ефективного відслідковування помилок, і швидкого внесення поправок, що значно полегшує процес розробки.

***Завершальний етап, хостинг.***

Завершальний етап розробки – є одним і найголовніших оскільки на даному етапі проводиться активний пошук помилок, оптимізація аплікації. І перевірка відповідності початковим вимогам.

Також на цьому етапі здійснюється так званий хостинг – тобто розміщення сайту в інтернеті.

Попередній хостинг здійснювався на ресурсі <http://www.smarterasp.net/> , який має бесплатний 60 денний період. Даний етап розробки неймовірно важливий, оскільки на ньому криється багато так званих підводних каменів.

Дуже часто навіть при успішній розробці локально, при виведенні сайту в інтернет з’являється велика кількість неочікуваних помилок, це зумовлено зміною середовища розгортання ресурсу, оскільки для успішного хостингу необхіно провести повну міграцію всіх даних і залежностей проекту на віддалений сервер. Тим не менше подолання цих проблем є невід’ємною частиною розробки будь якого проекту.

***Висновок***

Практика є джерелом накопичення практичного досвіду роботи по вибраній спеціальності та закріплення теоретичних знань. Вона відіграє важливу роль у підготовці спеціалістів високого рівня.

В ході проходження практики був здобутий необхідний практичний досвід в розробці веб аплікації, засвоєно нові трендові технології, а також значно розширені знання в галузі архітектури програмного забезпечення і втілення ефективних концепій у життя.