Company Name

Project Name

v.1

Spec Documentation

Last updated: 11 Feb 2023

Index

Introduction

Introduction

Blazor is a free, open-source framework for building modern, interactive web applications. It was developed by Microsoft as a part of the .NET platform, and it allows developers to use C# and .NET to create web applications that run in a web browser.

Blazor offers a unique approach to web development by using WebAssembly, a low-level binary format that runs in modern web browsers. This enables developers to use C#, instead of JavaScript, as the primary language for client-side development. With Blazor, developers can write code that runs directly in the browser, eliminating the need for a separate server-side component and allowing for a more streamlined development experience.

One of the key advantages of Blazor is that it leverages the full power of .NET and the C# language. This means that developers can use the same language, libraries, and tools for both client-side and server-side development, leading to increased productivity and a more streamlined development experience. Additionally, Blazor enables developers to reuse existing .NET code, libraries, and components, reducing the time and effort required to develop new applications.

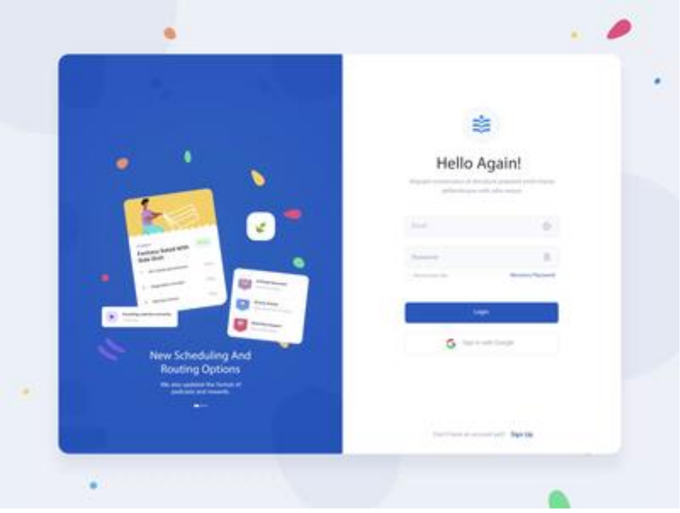
Blazor also provides a modern, component-based architecture that makes it easy to create reusable and modular components. These components can be easily shared and reused across projects, reducing development time and improving the overall quality of the code.

Blazor supports both client-side and server-side hosting models. The client-side model runs Blazor components directly in the browser using WebAssembly, while the server-side model runs Blazor components on the server and communicates with the client using SignalR, a real-time communication framework.

The server-side hosting model is ideal for scenarios where the client device has limited resources or slow network connections, as it offloads the processing to the server. The client-side hosting model, on the other hand, is ideal for scenarios where the client device has a fast network connection and ample resources, as it provides a more responsive and interactive user experience.

Workflow

1.1 Login



Description

A web login page is a type of login screen that is used on websites to authenticate users. It is a page that users are directed to when they need to log in to a website to access certain content or services.

APIs Used

1. /api/user/login [POST]

Stored Procedures

1. Login

   Input [JSON]

   ```
   {
       “Username”: “user”,
       “Password”:”1234abcA”
   }
   ```

   Output [JSON]

   ```
   {
       “ID”:1
       “Username”: “user”,
       “DisplayName”:”Admin User”
   }
   ```

Database & Tables

1. User

   Columns: [Username, Password]

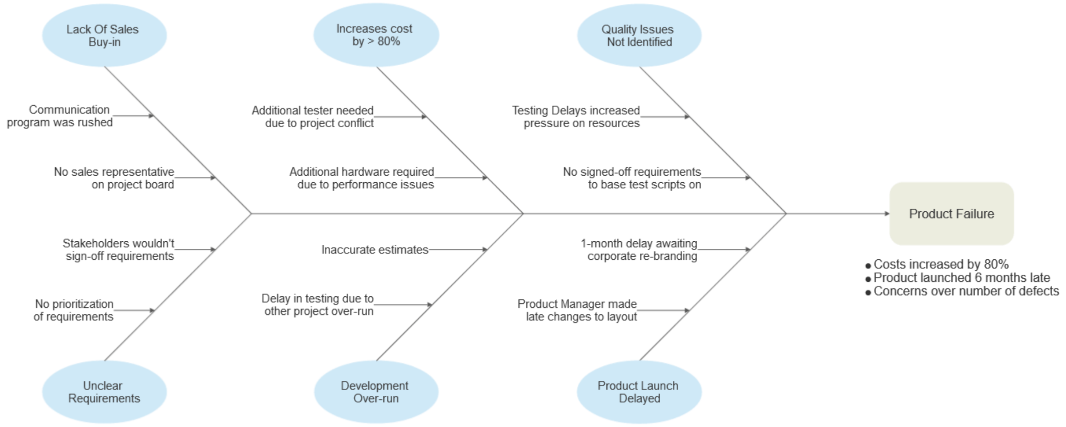
2. User Log

   Columns: [ID, Who, When, What, CreatedBy, CreatedTime]

Business logics behind

Business logic refers to the underlying rules and processes that govern the operation of a business. It encompasses the methods and algorithms that determine how a business operates and how it achieves its goals. Business logic can include everything from how a company processes orders to how it makes decisions about hiring, pricing, and marketing. The main purpose of business logic is to ensure that a business operates in a consistent and efficient manner, and that it is able to achieve its goals in the most effective way possible. Business logic is often implemented as software applications that automate various business processes, making it easier for businesses to manage their operations and stay on track.

Diagrams


Lack Of Sales Buy-in
Increases cost by > 80%
Quality Issues Not Identified
Communication program was rushed
Additional tester needed due to project conflict
Testing Delays increased pressure on resources
No sales representative on project board
Additional hardware required due to performance issues
No signed-off requirements to base test scripts on
Product Failure
Stakeholders wouldn't sign-off requirements
Inaccurate estimates
1-month delay awaiting corporate re-branding
No prioritization of requirements
Delay in testing due to other project over-run
Product Manager made late changes to layout
Unclear Requirements
Development Over-run
Product Launch Delayed

- Costs increased by 80%
- Product launched 6 months late
- Concerns over number of defects