Software Requirements Specification (SRS)

Project: hotel booking

Date: 2025-12-12

Prepared by: Abdelwhab 5587 & habib 5377

---

1. Introduction

1.1 Purpose

This document specifies the functional and non-functional requirements for the new project web application. It is intended for developers, testers, product owners, and stakeholders.

1.2 Scope

The application is a small Blazor web site containing pages: Home, Counter, Page1, Page2, Page3, Weather, and shared layout/navigation components[6]. The SRS covers user interactions, configuration, deployment constraints, and acceptance criteria.

1.3 Definitions, acronyms, abbreviations

- **Blazor**: Microsoft framework used by the app.

- **App**: HotelBooking web application.

---

2. Overall Description

2.1 Product perspective

The app is a single-site Blazor web app served by ASP.NET Core. It uses the MudBlazor component library and is configured for interactive server render mode. Files are organized under the HotelBookingBlazor/ folder.

2.2 User classes and characteristics

- **Visitor/User**: Browses pages, views weather, and increments a counter.

- **Administrator**: Manages configuration and deployment.

2.3 Operating environment

- **Client**: Modern browsers (Chrome, Edge, Firefox) on desktop and mobile.

- **Server**: ASP.NET Core on Windows/Linux hosting supporting .NET 9.0 runtime.

2.4 Design and implementation constraints

- Uses Blazor components and server/static files in wwwroot/.

- Configuration via appsettings.json and appsettings.Development.json.

2.5 Assumptions and dependencies

- If an external weather API is used, an API key and reachable endpoint are required.

- Browser JavaScript must be enabled for interactive components.

3. Functional Requirements (FR)

ID	Page / Component	Description	Acceptance Criteria
FR-1	Home Page	Displays welcome content and navigation shortcuts via Home.razor.	Loads in < 2s; links open destination pages.
FR-2	Counter Page	Allows user to increment a counter via an @onclick event handler.	Clicking increment updates the value immediately; button is keyboard accessible.
FR-3	Page 1, 2, 3	Placeholder pages with specific content and navigation.	Pages load and show expected content.
FR-4	Weather Page	Generates sample forecast data on initialization with a loading indicator.	Shows loading spinner then renders 5 forecast rows.
FR-5	Shared Layout	MainLayout.razor provides header/footer; NavMenu.razor handles navigation.	Navigation links are reachable and layout is consistent.

4. External Interface Requirements

4.1 User Interface

- **Browsers**: Support latest two major versions of Chrome, Edge, Firefox.

- **Responsive behavior**: Layout scales to common mobile and desktop widths.

---

5. Non-Functional Requirements (NFR)

- **NFR-1: Performance**: First meaningful paint within 2s; page navigation under 1s when cached.

- **NFR-2: Scalability**: Host must support concurrent visitors typical for the deployment environment.

- **NFR-3: Reliability**: Application should target 99.5% uptime when deployed with standard hosting.

- **NFR-4: Maintainability**: Code should follow component patterns and maintain separation between UI and data access.

---

6. Data Requirements

- **DR-1: Settings storage**: appsettings.json contains endpoints and toggles; secrets stored securely in production.

- **DR-2: Transient UI state**: Counter state is client-side only unless persistence is added.

- **DR-3: Logging**: Use ASP.NET logging for errors and key events.

---

7. Use Cases

- **UC-1: Browse Home**: Visitor navigates to home URL, page renders, and user sees navigation links.

- **UC-2: Increment Counter**: Visitor opens Counter page and clicks increment; value increases instantly.

- **UC-3: View Weather**: Visitor opens Weather page; app fetches/generates data and displays forecasts.

---

8. Acceptance Criteria (AC) Traceability

- **AC-1**: All nav links reach correct pages (FR-1, FR-5).

- **AC-2**: Incrementing updates number; test keyboard activation (FR-2).

- **AC-3**: Weather page displays >0 items with correct configuration (FR-4).

- **AC-4**: Weather page shows fallback message on network error (FR-4).

9. Deployment & Operational Notes

- **Deployment**: Build with .NET 9.0; host on cloud App Service.

- **Monitoring**: Configure application logs and uptime monitor.