**2. Documento Tecnico – Home Page (Sviluppo 2)**

**1. Architettura e Struttura del Progetto**

**Tecnologie di Base:**

* **Framework: Angular (solo front-end).**
* **Stile e Layout:**
  + **CSS/SCSS con l’utilizzo di Flexbox o CSS Grid per il layout responsive.**
  + **Utilizzo di Angular Material o componenti custom per l’interfaccia utente.**

**2. Componenti e Moduli Angular**

**Componenti Principali:**

* **HeaderComponent:**
  + **Responsabilità:**
    - **Visualizzare il logo "Parthenope Banking".**
    - **Gestire il menu di navigazione con le voci principali.**
* **HomeContentComponent:**
  + **Responsabilità:**
    - **Visualizzare il riepilogo dei dati finanziari (saldo, ultime operazioni, notifiche).**
    - **Contenere widget interattivi per l’accesso rapido ad altre funzioni.**
* **WidgetComponent (opzionale):**
  + **Responsabilità:**
    - **Componente riutilizzabile per visualizzare dati sintetici e aggiornamenti in tempo reale.**

**3. Struttura del Layout e Routing**

* **Layout della Pagina:**
  + **Header fisso (riutilizzabile in tutte le pagine interne).**
  + **Area contenuto principale che si aggiorna dinamicamente.**
* **Routing:**
  + **Configurazione delle rotte per le diverse sezioni (es. /home, /conti, /bonifici, /profilo).**
  + **Utilizzo di Angular Router per gestire la navigazione e il caricamento dinamico dei componenti.**

**4. Stile e UI**

**Palette e Temi:**

* **Colori:**
  + **Azzurro (#428fdd) per header, pulsanti, link e altri elementi interattivi.**
  + **Bianco per lo sfondo della pagina e dei contenuti.**
* **Design Responsive:**
  + **Media query e layout a griglia per garantire un’esperienza ottimale su ogni dispositivo.**
* **Componenti UI:**
  + **Utilizzo di librerie (Angular Material o simili) per componenti standard come toolbar, cards e list.**

**5. Gestione dello Stato e Interattività**

* **Servizi Angular:**
  + **Creazione di un HomeService per simulare il recupero dei dati (es. saldo, movimenti, notifiche) tramite dati mock.**
* **Binding Dati:**
  + **Uso di data binding e direttive Angular per aggiornare dinamicamente l’interfaccia in risposta ai dati del servizio.**
* **Animazioni e Transizioni:**
  + **Implementazione di transizioni CSS per migliorare l’esperienza utente durante il passaggio tra sezioni e il caricamento dei widget.**

**6. Scalabilità e Manutenibilità**

* **Componentizzazione:**
  + **Strutturare l’interfaccia in componenti modulari e riutilizzabili per facilitare la manutenzione e l’espansione futura.**
* **Separazione delle Preoccupazioni:**
  + **Distinzione chiara tra logica di presentazione (componenti) e logica di business (servizi).**
* **Documentazione Interna:**
  + **Commenti e documentazione nel codice per agevolare futuri sviluppatori nella comprensione del flusso e delle logiche implementate.**