**Project Design Phase**

**Solution Architecture**

|  |  |
| --- | --- |
| Date | 25 June 2025 |
| Team ID | LTVIP2025TMID59290 |
| Project Name | FlightFinder: Navigating Your Air Travel Options |
| Maximum Marks | 4 Marks |

**Solution Architecture:**

**Solution architecture is a strategic approach that connects the Flight Finder business needs—streamlined flight booking, tracking, and user experience—with the most suitable and scalable technical implementation.**

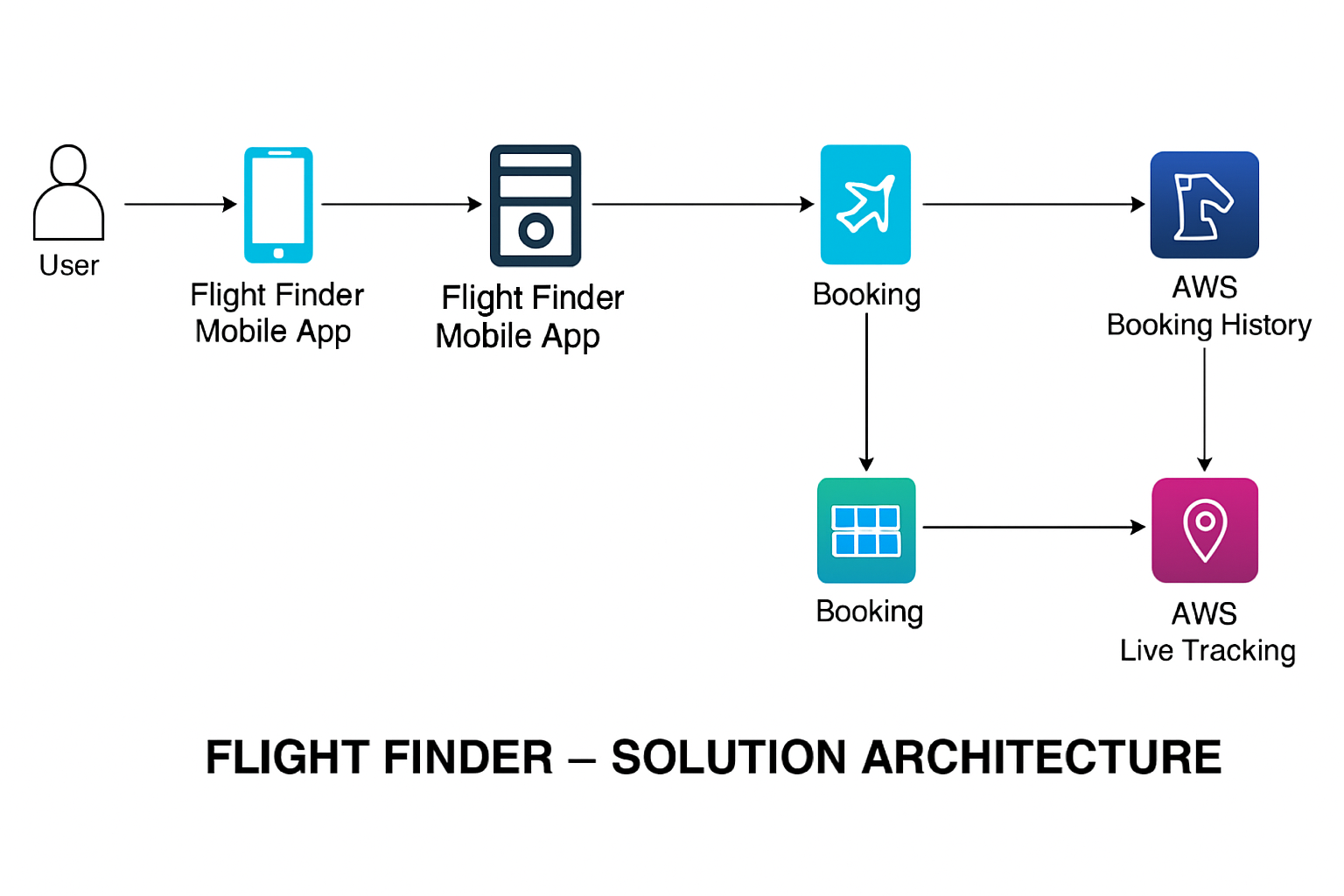
**Goals of the Solution Architecture**

1. **Find the Best Tech Solution  
   Identify and implement a scalable, user-centric, and secure flight booking platform accessible via web and mobile.**
2. **Describe System Characteristics to Stakeholders  
   Clearly present the system’s:**
   * **User workflows (registration, booking, tracking)**
   * **Component interactions (frontend, backend, databases, third-party APIs)**
   * **Cloud services used (e.g., AWS, Firebase, or Supabase)**
   * **Security and performance attributes**
3. **Define Features, Phases, and Requirements**
   * **Core Features:**
     + **User registration and login (via form or social auth)**
     + **Flight search with filters**
     + **Booking and cancellation**
     + **Booking history and live flight tracking**
   * **Phases:**
     + **MVP: Registration, login, search, booking**
     + **Phase 2: History, tracking, admin panel, notifications**
4. **Provide Technical Specifications  
   The solution is implemented using:**
   * **Frontend: React/Next.js (for Web), React Native or Flutter (for Mobile)**
   * **Backend: Node.js with Express.js or Next API routes**
   * **Database: PostgreSQL (via Supabase or hosted DB)**
   * **Cloud Services: AWS for file storage, email, analytics, and tracking**
   * **Authentication: Supabase Auth or Firebase Auth with OTP/Social Login**
   * **DevOps: CI/CD pipeline with GitHub Actions & deployment via Vercel**

**Architecture Blueprint Highlights**

* **User Journey:**
  + **User interacts via web/mobile UI**
  + **Requests sent to backend APIs**
  + **Flight data fetched from database or third-party APIs (Skyscanner, Amadeus, etc.)**
  + **Booking info stored securely**
  + **Admin dashboards for flight management**
  + **Live status fetched and displayed in real-time**
* **Security & Compliance:**
  + **End-to-end encryption**
  + **Role-based access control**
  + **Secure data storage and logging**
  + **Compliance with data privacy laws (GDPR)**

**Flight Finder – Solution Architecture Diagram:**



*Figure 1: Architecture and data flow of the voice patient diary sample application*

**Reference:** [**https://aws.amazon.com/blogs/industries/voice-applications-in-clinical-research-powered-by-ai-on-aws-part-1-architecture-and-design-considerations/**](https://aws.amazon.com/blogs/industries/voice-applications-in-clinical-research-powered-by-ai-on-aws-part-1-architecture-and-design-considerations/)