**Detailed Requirements for GameTime**

**Functional Requirements**

**1. Key Functions and Features**

1. **User Registration and Login:**
   * Users can sign up using email, Facebook, or Apple accounts.
   * Secure login and password reset options.
2. **Personalized Player Profiles:**
   * Players complete a detailed questionnaire about their skill level, position, age range, fitness level, and preferred playing surface.
3. **Game Creation and Management:**
   * Organizers can create football games with details like time, location, number of players, and level of play.
   * Option to manage player cancellations and find substitutes in real-time.
4. **Matchmaking and Team Balancing:**
   * Players are matched to games based on their profile data.
   * Automatic team creation ensuring balanced skill levels and positions.
5. **Community Features:**
   * Chat and notifications for game updates.
   * Ability to invite friends to games.
6. **Dynamic Updates:**
   * Notifications for field availability, weather conditions, or last-minute changes.
7. **Location and Facility Integration:**
   * Interactive maps for locating nearby fields and indicating surface types (grass, asphalt).
8. **Feedback and Ratings:**
   * Players can rate games and provide feedback on the experience.

**2. Use Cases**

**Use Case 1: Joining a Football Game**

* **Pre-Conditions:** The user is logged in and has a completed profile.
* **Basic Flow:**
  1. User searches for games within a specified radius.
  2. The system suggests games based on the user’s profile.
  3. User selects a game and confirms participation.
  4. Organizer receives a notification about the new participant.
* **Post-Conditions:** User is added to the game roster.
* **Alternate Flow:** If no suitable game is found, the system suggests creating a new game.

**Use Case 2: Handling Player Cancellations**

* **Pre-Conditions:** A game is scheduled, and a player cancels their participation.
* **Basic Flow:**
  1. Player notifies the system of their cancellation.
  2. The system alerts the organizer and searches for substitutes from the waitlist or nearby users.
  3. Replacement player confirms their participation.
* **Post-Conditions:** Game roster is updated with the replacement player.

**3. User Stories**

1. **As a Player**, I want to join games that match my skill level and location so that I can enjoy a balanced and convenient football experience.
   * **Acceptance Criteria:** Games suggested must be within 10km of my location and match my preferred skill level.
2. **As an Organizer**, I want to manage player cancellations easily so that games are not disrupted.
   * **Acceptance Criteria:** Substitutes are found and confirmed within 15 minutes of a cancellation.
3. **As a Player**, I want to see detailed game information (time, location, surface) so that I can decide whether to join.
   * **Acceptance Criteria:** All game details are displayed clearly on the game page.

**4. Identify Stakeholders**

The primary stakeholders involved in the **GameTime** project include:

1. **End-Users:**
   * **Players**: Individuals looking to join casual football games in their local community.
   * **Organizers**: Users managing football events and ensuring a smooth game experience.
2. **Project Managers:**
   * Responsible for overseeing the development and implementation of the platform.
3. **Developers:**
   * Software engineers and designers working on the platform’s functionality and user interface.
4. **Local Business Owners:**
   * Operators of sports facilities (e.g., football fields) who can partner with GameTime to host games.

**5. Conduct Stakeholder Interviews / Workshops / Surveys**

**Interviews:**

* Conduct one-on-one interviews with players and organizers to gather insights into their challenges and expectations.
* **Sample Questions:**
  1. "What are the biggest challenges you face when organizing football games?"
  2. "What features would make joining games easier for you?"
  3. "How important is real-time communication during game planning?"

**Workshops:**

* Host focus groups with potential users to discuss desired functionalities and test early prototypes.
* Include discussions about balancing teams, managing cancellations, and integrating weather updates.

**Surveys:**

* Distribute surveys to a larger audience to collect quantitative data on preferences.
* **Sample Survey Questions:**
  + "How often do you play casual football games?"
  + "Would you prefer an app that matches players based on skill level?"
  + "What additional features would improve your football experience?"

**6. Review Existing Documentation**

**Existing Sources to Analyze:**

1. **Business Plans:** Review initial GameTime business objectives to align the development process with organizational goals.
2. **Competitor Analysis:**
   * Study reports on platforms like Hobiz and Meetup to identify strengths and weaknesses.
   * Extract insights into features that are missing or underdeveloped in competing platforms.
3. **Community Data:** Examine reports or studies on amateur football trends, including participation rates and common challenges.

**7. Identify Functional and Non-Functional Requirements**

**Prioritization of Requirements:**

* **Critical Requirements:**
  1. User registration and profile creation.
  2. Game creation and matchmaking based on player preferences and skill levels.
  3. Real-time notifications for game updates and cancellations.
  4. GPS integration for finding nearby fields.
* **Moderate Requirements:**
  1. Team balancing algorithms to ensure fair play.
  2. Weather API integration for suggesting alternative plans.
  3. Interactive maps with field information (surface type, availability).
* **Low-Priority Requirements:**
  1. Group chat for players within games.
  2. Invitation features for friends.

**Non- Functional, Technological and Architectural Requirements**

**1. System Architecture**

* **Architecture Style:**
  + Microservices architecture to maintain modularity and flexibility, but we may consider simpler solutions if time or resources are limited.
* **Key Components:**
  + **Frontend and Backend:** The platform will run on Flutter, supporting both mobile and web versions.
  + **Database:** PostgreSQL for structured data and Redis for caching real-time updates.
  + **APIs:** Integration with Google Maps for location services and OpenWeatherMap for weather updates.

**\*\* Changes may occur \*\***

**2. Scalability**

* **Vertical Scaling:** Increase server resources during peak times.
* **Horizontal Scaling:** Consider adding more servers as the platform grows, but we aim to minimize the need for horizontal scaling in the initial phase.

**3. Performance**

* **Response Time:** The platform will respond to user actions as fast as could.
* **Throughput:** We aim to support up to 100 concurrent users initially.

**4. Reliability and Availability**

* **Uptime:** The platform will have at least 99% uptime, with manual or automatic failover mechanisms for basic fault recovery.
* **Redundancy:** Basic server redundancy will be implemented, but a fully advanced backup system may not be necessary at this stage.

**5. Security**

* **Authentication:** Login By User name and password.

**6. Data Architecture**

* **Database Design:**
  + Tables for user profiles, games, locations, and ratings.
* **Data Flow:**
  + User data will be stored in PostgreSQL, with Redis used for real-time updates (e.g., game changes, weather).

**7. Integration**

* **External Systems:**
  + Integration with Google Maps for GPS-based services.
  + Integration with Weather app for real-time weather updates.

**8. Maintainability**

* **Code Maintainability:** The platform will be designed for easy updates and future additions, using clear and simple code structures with sufficient documentation for future development.
* **Testing:** Basic functional testing will be performed on each component, but advanced automated testing solutions may not be implemented at this stage.

**9. Usability**

* **User Interface (UI):** The platform will be designed to be user-friendly, with a simple and intuitive interface to avoid overwhelming users.
* **User Experience (UX):** The goal is to provide a smooth user experience, making it easy for users to find and join games quickly.