**1. Requirements Analysis**

**Objectives:**

* Identify the functionalities the system must perform.
* Define technical and non-technical needs.
* Ensure the system aligns with the project’s goals.

**Key Sections:**

1. **Functional Requirements:**
   * Allow users to input tourist attractions and governorates.
   * Search for tourist attractions based on the governorate.
   * Display details of tourist spots (name, address, image).
   * Include additional features, such as user ratings or historical details.
2. **Non-Functional Requirements:**
   * User-friendly interface.
   * Fast data retrieval.
   * Multi-user support.
   * Compatibility with multiple devices (mobile, tablet, desktop).
3. **Data Requirements:**
   * A data model containing:
     + Names of tourist attractions.
     + Addresses (detailed location and governorate).
     + Images of the attractions.
     + Additional details (category: historical, natural, cultural).
4. **Technical Requirements:**
   * Database (e.g., SQLite, MySQL).
   * Frontend interface (web-based or desktop/mobile application).
   * Programming language (Python with Pandas and Streamlit or Django).
5. **User Requirements:**
   * **Tourist**: Search for tourist attractions.
   * **Administrator**: Update data and add new attractions.

**2. Fact Gathering**

**Methods for Fact Gathering:**

1. **Interviews:**
   * Conduct interviews with tourism experts to understand what tourists need.
   * Ask site managers about key information tourists inquire about.
2. **Surveys:**
   * Create surveys for tourists to understand their preferences:
     + How do you select a tourist destination?
     + Do you prefer seeing images and locations before deciding?
3. **Document Analysis:**
   * Review the current Excel dataset to determine if it’s complete or requires enhancement.
   * Identify missing columns, such as operating hours or entry fees.
4. **Observation:**
   * Analyze similar tourism platforms (e.g., TripAdvisor) to identify useful features.

**3. Fact Recording**

|  |
| --- |
| **Sample Documentation:** |
| | **Category** | **Description** | | --- | --- | | **Target Users** | Tourists, tourism site administrators. | | **Data Sources** | Current Excel file, interviews with tourism experts. | | **Functional Needs** | Search, display images and addresses, provide additional information. | | **Technical Needs** | Python, Pandas, Streamlit or Django framework, SQLite database. | | **Data Enhancements** | Add columns like working hours, entry fees, and visitor ratings. | |

**4. Outcome of Analysis and Fact Gathering**

* After gathering facts and analyzing requirements, you will have a clear understanding of the system's structure.
* You can begin developing a **prototype** to test the functional requirements.

**Functional Requirements**

These are the core functionalities the system must perform to meet user needs:

1. **Search Functionality:**
   * Allow users to search for tourist attractions based on the governorate name.
   * Provide filters for users to narrow down results (e.g., by type: historical, cultural, natural).
2. **Data Presentation:**
   * Display details of tourist spots, including:
     + Name.
     + Governorate.
     + Address.
     + Image.
3. **User Interaction:**
   * Allow users to view tourist spot information in a user-friendly format.
   * Optionally, allow users to rate or review tourist spots.
4. **Data Management:**
   * Provide administrative tools for updating, adding, or removing tourist attractions.
   * Validate data inputs to ensure consistency and accuracy.
5. **Multilingual Support (Optional):**
   * Display information in multiple languages for international tourists.
6. **Data Export (Optional):**
   * Allow users to download the list of tourist spots in formats like PDF or Excel.

**Non-Functional Requirements**

These describe the quality attributes, performance, and constraints of the system:

1. **Performance:**
   * The system should respond to search queries within 2 seconds.
   * Handle up to 100 concurrent users without significant slowdowns.
2. **Scalability:**
   * The system must support an increasing number of tourist spots and users over time.
3. **Usability:**
   * Provide a simple and intuitive user interface.
   * Ensure accessibility for users with disabilities (e.g., screen readers).
4. **Reliability:**
   * Ensure that the system is available 99.9% of the time (minimal downtime).
   * Provide a mechanism to recover from crashes or data corruption.
5. **Security:**
   * Protect sensitive data (e.g., admin login credentials) with encryption.
   * Prevent unauthorized data modifications.
6. **Compatibility:**
   * The system should be accessible on various devices (desktops, tablets, smartphones).
   * Support major browsers (Chrome, Safari, Firefox, Edge).
7. **Maintainability:**
   * Use modular and clean code for easier updates and bug fixes.
   * Provide clear documentation for future developers.
8. **Data Integrity:**
   * Ensure no duplicate or inconsistent data entries in the system.
9. **Localization:**
   * Provide support for displaying dates, currencies, and addresses in local formats.
10. **Regulatory Compliance:**
    * Comply with data protection regulations, such as GDPR, if applicable.

**Context Diagram for Tourism Management System (TMS)**

**Description: A Context Diagram illustrates the system as a central process interacting with external entities such as tourists, administrators, external data sources, and output channels. It highlights the interactions with inputs and outputs for each entity without going into internal processes.**

**Entities and Interactions:**

1. **Tourists:**
   * **Input: Number of days for the trip, total available budget, search criteria (e.g., type of attraction, location).**
   * **Output: List of recommended attractions, including names, addresses, costs, ratings, opening hours, best times to visit, and images.**
2. **Administrators:**
   * **Input: Manage data of tourist attractions (add, edit, delete) via the system interface. This includes adding details like name, address, description, cost, type, image, and opening hours.**
   * **Output: Updated system data, including new or modified tourist attraction records.**
3. **External Data Source:**
   * **Input: Excel file or other data sources containing tourist attraction information.**
   * **Output: Data imported and validated into the system, which includes details like attraction types, costs, ratings, descriptions, and geographic coordinates (latitude and longitude).**
4. **Output Channels:**
   * **Input: Requests for viewing/searching or exporting data (such as PDF or Excel).**
   * **Output: Data displayed via the user interface (web/mobile), such as a trip plan or interactive map with attraction locations, cost, and rating details.**

**Detailed Diagram Description:**

1. **Tourists' Interaction with the System:**
   * **Input: Tourists input the number of days for the trip, budget, and preferences (such as attraction type or location) into the system.**
   * **Output: The system recommends a trip plan with a list of tourist attractions within the daily budget. It displays detailed information like cost, rating, opening hours, description, and links for booking or further details.**
2. **Administrators' Interaction with the System:**
   * **Input: Administrators add or update attraction data in the system, including basic details (name, type, location) and additional information (cost, rating, best times to visit).**
   * **Output: Updated system data, ensuring the list of tourist attractions is current and consistent for users to access.**
3. **External Data Source Interaction:**
   * **Input: The system receives data from external sources (such as an Excel file or other structured datasets) containing information about tourist attractions.**
   * **Output: Imported and processed data is validated, integrated into the system, and made available for use in trip planning.**
4. **Output Channels Interaction:**
   * **Input: Users or administrators request data (e.g., trip plans, search results).**
   * **Output: The system responds by displaying data through a web/mobile interface, providing real-time search results or exporting data in different formats (e.g., PDF, Excel). Additionally, an interactive map with tourist attraction locations is generated.**

**Shehap Elakhder
**