1. **Functional Requirements:**
   - Define the specific features and functions the ATS should support, such as job posting, application submission, candidate evaluation, interview scheduling, and reporting.
2. **Technology Stack:**
   - Choose the appropriate technologies for each component of the system. For example, you might use a web application framework like Ruby on Rails, Django, or Node.js for the frontend, and a database like PostgreSQL or MySQL for data storage.
3. **Components of the ATS:**

   a. **User Interface (UI):**
   - Design a user-friendly web-based interface for both candidates and recruiters.
   - Implement responsive design to support various devices (desktop, mobile).
   - Ensure accessibility and usability.

   b. **Application Database:**
   - Set up a relational database to store all application data, including job listings, candidate profiles, and application histories.
   - Ensure data security and compliance with data protection regulations (e.g., GDPR).

   c. **User Authentication and Authorization:**
   - Implement secure user authentication and authorization mechanisms to control access to the system.
   - Support role-based access control to differentiate between recruiters, hiring managers, and candidates.

   d. **Search and Matching Engine:**
   - Develop algorithms for matching job requirements with candidate profiles.
   - Implement search functionality for easy retrieval of job listings and candidate profiles.

   e. **Workflow and Notifications:**
   - Create a workflow engine to support the hiring process stages (e.g., screening, interviewing, offer, onboarding).
   - Implement email notifications and alerts to keep users informed about application status changes.

   f. **Reporting and Analytics:**
   - Integrate reporting tools to allow recruiters and managers to track application metrics, such as time-to-hire, source of candidates, and diversity statistics.
   - Implement data analytics to help in decision-making and process improvement.

g. **Integration with External Systems:**
   - Enable integration with external job boards, HR systems, and background screening services.
   - Use APIs and webhooks to exchange data with third-party platforms.

h. **Scalability and Performance:**
   - Design the architecture to be scalable to handle a growing number of job listings, candidates, and users.
   - Implement caching mechanisms and load balancing to ensure optimal performance.

i. **Security and Compliance:**
   - Implement security measures, including data encryption, penetration testing, and regular security audits.
   - Ensure compliance with relevant regulations like GDPR, HIPAA (if applicable), and industry-specific standards.

j. **Data Backup and Recovery:**
   - Set up regular data backups and disaster recovery mechanisms to prevent data loss.

4. **Deployment Model:**
   - Decide whether to deploy the ATS on-premises, in a private cloud, or leverage a public cloud infrastructure like AWS, Azure, or Google Cloud.
5. **Testing and Quality Assurance:**
   - Plan for thorough testing of the system to identify and resolve bugs and issues.
   - Implement automated testing and continuous integration to streamline development and deployment.
6. **Documentation and Training:**
   - Create comprehensive documentation for system administrators and end-users.
   - Provide training for HR personnel and recruiters on how to use the ATS effectively.
7. **Monitoring and Maintenance:**
   - Implement monitoring tools to track system performance and user activity.
   - Set up regular maintenance tasks, including software updates, security patches, and database optimization.
8. **Scalability and Future-proofing:**
   - Design the architecture to accommodate future enhancements and requirements as the organization grows and evolves.
9. **Cost Considerations:**
   - Estimate the cost associated with the infrastructure, development, maintenance, and licensing.
10. **Data Privacy and GDPR Compliance:**

- Ensure that the system complies with data privacy regulations, especially GDPR, by implementing features such as data access requests and consent management.

Once you have designed the architecture, you can move on to the development, testing, and deployment phases of the project. Keep in mind that effective communication and collaboration between developers, designers, and stakeholders are crucial for the successful implementation of the ATS.