**Functional requirements**

**User Authentication and Authorisation**

* **Login/Logout**: Users should be able to login and logout securely.
* **Role-based Access control (RBAC):**
  + **End-Users:** Can upload records, view predictions, and provide feedback.
  + **AI Engineers:** Can upload new ML models and access interaction logs. (refine models)?
  + **Administrators:** Can monitor system activity, manage users and monitor level of activity for MLAAS
  + **Finance Teams:** Can generate billing and invoices based on user activity.

**Dashboard interfaces**

* **End-User Dashboard:**
  + Upload records (text, image, video, audio).
  + View predictions from the ML model.
  + Provide feedback on predictions.
* **AI Engineer Dashboard:**
  + Upload new ML models
  + Access user interaction logs
  + Refine ML models?
* **Administrator Dashboard:**
  + Monitor system activity
  + Monitor user activity
  + Manage user accounts
* **Finance Dashboard:**
  + Generate billing and invoices

**GDPR Compliance**

* **Data anonymisation:** ensure all personal data is anonymised where necessary**.**
* **Explainability:** provide explanations for the model’s predictions to ensure transparency**.**
* **Fairness and Accountability:** Monitor model predictions for bias.
* **Data Protection:** Implement measures to protect personal data in compliance with GDPR

**Non-Functional Requirements**

**Performance**

* **Response Time:** The system should respond to user requests within 2 seconds.
* **Scalability:** The system should handle up to 10,000 concurrent users.
* **Throughput:** The system should process at least 100 requests per second.

**Security**

* **Data Encryption:** All sensitive data should be encrypted in transit and at rest.
* **Authentication:** Implement secure authentication mechanisms (e.g., OAuth, JWT).
* **Authorisation:** Ensure role-based access control is strictly enforced.
* **Audit Logs:** Maintain audit logs for all critical actions.

**Reliability**

* **Error Handling:** The system should handle errors and provide meaningful error messages.

**Usability**

* **User Interface:** The dashboard should be intuitive and easy to use.
* **Accessibility:** The system should be accessible to users with disabilities.
* **Documentation:** Provide comprehensive documentation for all user roles.

**Maintainability**

* **Code Quality:** Follow best practices for code quality and maintainability.
* **Modularity:** Design the system modularly to facilitate future updates.
* **Testing:** Implement automated testing for all critical components.

**Jira tasks**

**User Authentication and Authorisation**

* **Task 1.1:** Implement Login/Logout functionality. (sessions must be used)
* **Task 1.2:** Implement Role-Based Access Control (RBAC).
* **Task 1.3:** Create user roles (End-User, AI Engineer, Administrator, Finance).