**Requirement Analysis for Breast Cancer Detection Website**

**1. Functional Requirements**

These are features and functionalities that the system must provide:

**Frontend Requirements:**

* **User Authentication:**
  + Users can register and log in using secure credentials (if react – JWT, if Python – Authlib).
  + Password reset functionality.
* **Input Submission:**
  + Users can upload mammogram images (JPEG/PNG format).
  + Users can input personal data (e.g., age, medical history).
* **Dashboard:**
  + Display diagnostic results (from CNN and SVM models).
  + Allow users to download or view their diagnostic reports.
  + Display the history of past reports.
* **Responsive Design:**
  + The website must be responsive and accessible on mobile, tablet, and desktop devices.

**Backend Requirements:**

* **ML Model Integration:**
  + Process uploaded mammogram images using the CNN model.
  + Process structured data inputs using the SVM model.
  + Combine results for a final decision.
* **Report Generation:**
  + Generate diagnostic reports in PDF format.
  + Store reports and make them available for users to download.
* **Data Handling:**
  + Securely store user information, diagnostic data, and uploaded images.
  + Handle large file uploads and prevent data loss during processing.
* **API Endpoints:**
  + Provide RESTful APIs for frontend-backend communication.

**Database Requirements:**

* Store user details, mammogram image links, diagnostic data, and reports.
* Ensure encryption for sensitive user data like passwords.
* Maintain relationships between users and their reports.

**Storage Requirements:**

* Efficiently store mammogram images.
* Ensure reports are stored and easily accessible for download.

**2. Non-Functional Requirements**

These are qualities and constraints of the system:

**Performance:**

* The system must process images and data inputs in under 10 seconds for a single request.
* The website should handle concurrent users (scalable to at least 100 simultaneous users).

**Scalability:**

* The system must support increased usage with cloud storage and additional compute resources (e.g., for model inference).

**Reliability:**

* Ensure 99.9% uptime.
* Regular database backups to prevent data loss.

**Security:**

* Use HTTPS to encrypt all communications.
* Passwords must be hashed using a secure algorithm (e.g., bcrypt).
* Limit file uploads to verified users to prevent misuse.

**Usability:**

* Intuitive and easy-to-use UI/UX for non-technical users.
* Error messages and success notifications for better user experience.

**Maintainability:**

* Use modular code with proper documentation.
* Employ version control (e.g., Git) for development.

**Compliance:**

* Adhere to HIPAA (Health Insurance Portability and Accountability Act) or equivalent standards for medical data privacy.
* Comply with GDPR for users in the EU.

**3. Technical Requirements**

**Frontend:**

* **Languages/Frameworks:** HTML, CSS, JavaScript (React.js optional for SPA functionality)
* **Tools:** Bootstrap or Tailwind CSS for styling, Axios for API calls

**Backend:**

* **Languages/Frameworks:** Python (Flask/Django)
* **Libraries:**
  + Flask-RESTful/Django REST Framework for APIs
  + Joblib for ML model serialization
  + PDF generation tools like WeasyPrint or FPDF

**Machine Learning:**

* **CNN Model:**
  + Framework: TensorFlow or PyTorch
  + Task: Image classification
* **SVM Model:**
  + Framework: Scikit-learn
  + Task: Data classification

**Database:**

* SQLite for development, PostgreSQL or MySQL for production
* ORM: SQLAlchemy (Flask) or Django ORM

**Storage:**

* Local storage for development
* Cloud storage (e.g., AWS S3, Google Cloud) for production

**Deployment:**

* Cloud hosting services like AWS, Google Cloud, or Azure
* Docker for containerization
* CI/CD tools like GitHub Actions or Jenkins

**4. User Stories**

1. **As a user**, I want to register and log in so that I can securely access the website.
2. **As a user**, I want to upload a mammogram image so that I can get diagnostic results.
3. **As a user**, I want to input personal data so that it can be analyzed alongside the image.
4. **As a user**, I want to view and download diagnostic reports so that I can share them with my doctor.
5. **As an admin**, I want to monitor the system to ensure smooth operation and manage user queries.

**5. Risks and Challenges**

1. **Data Privacy:** Protecting sensitive medical data is critical.
   * Mitigation: Use encryption and follow compliance guidelines.
2. **Performance Bottlenecks:** Processing large mammogram images may slow the system.
   * Mitigation: Optimize ML models and use GPUs for inference.
3. **Scalability:** Handling high user traffic and storage requirements.
   * Mitigation: Use scalable cloud infrastructure.
4. **Model Accuracy:** Ensuring ML models provide accurate predictions.
   * Mitigation: Test models with diverse datasets and fine-tune hyperparameters.
5. **File Uploads:** Large image files might cause delays or crashes.
   * Mitigation: Set file size limits and use asynchronous processing.