CS13 Makhimeter 3.0	Anusha Khan 21B-032-CS AbdulRauf Siddiqui 21B-046-CS Areeba Aamir 21B-133-CS Areesha Mallick 21B-208-CS Syed Hadeed Khalid 21B-217-CS

Feature List

Features are defined in four areas (input, output, process, and nonfunctional requirements), each statement can be written as

As a < type of user >, I want < some goal > so that < some reason >.

With acceptance criteria

For help : https://www.mountaingoatsoftware.com/agile/user-stories

https://www.altexsoft.com/blog/business/acceptance-criteria-purposes-formats-and-best-practices/

. Inputs

Inputs are essential data sources and interactions that Makhimeter relies on to perform automated analysis of *Drosophila melanogaster* images. These inputs come from users, external sources, and internal systems.

1.1 User Inputs

- **Image Data Entry**

User Story: As a Researcher or Lab Technician, I want to upload images of *Drosophila melanogaster* so that the system can analyze genetic traits and behavioral patterns.
Acceptance Criteria: The system should provide a user-friendly interface for image uploads with real-time feedback on the image format and resolution.

- **Parameter Configuration**

User Story: As a Bioinformatics Specialist, I want to configure analysis parameters (e.g., segmentation thresholds, model settings) to customize the image analysis for specific research needs.
Acceptance Criteria: The system should allow parameter configuration through a visual interface, providing options for default and custom settings.

- **Genotype Classification Input**

User Story: As a Researcher, I want to input movement video data to predict the fly's genotype based on its flight trajectory.
Acceptance Criteria: The system should accept structured input data for flight analysis and apply the trained model to predict genotypes accurately.

1.2 External Inputs

Image Datasets from External Sources

- **User Story:** As a Researcher, I want to integrate external datasets of *Drosophila* images from the DOW Fly Lab to enhance model training and validation.
- **Acceptance Criteria:** The system should support importing image datasets from external sources in common formats (e.g., JPEG, PNG) and validate them for consistency and quality.

1.3 Internal Inputs

- **Internal Database Storage**
  User Story: As a System Administrator, I want to store preprocessed and augmented images in an internal database to maintain a consistent and accessible data repository for model training.
  Acceptance Criteria: The database should support efficient data storage, retrieval, and backup mechanisms to ensure data integrity and accessibility.
- **Data Processing Parameters**

User Story: As a Bioinformatics Specialist, I want to adjust data parameters (e.g., segmentation threshold values, image dimensions) to optimize the model's input data.

Acceptance Criteria: The system should allow dynamic adjustment of data processing parameters with minimal disruption to ongoing analyses.

Process: Steps of the Project

1. **Data Collection and Preprocessing**
    - **Description:** Gather datasets from researchers and external sources, perform data augmentation, and preprocess images to create a consistent and diverse dataset.
    - **Outcome:** A high-quality dataset ready for training and analysis, stored in the internal database.
2. **Model Development and Training**
    - **Description:** Design and train Convolutional Neural Network (CNN) models for tasks such as wing segmentation, brain scan classification, and fly trajectory analysis.
    - **Outcome:** Trained models that are accurate and reliable for analyzing *Drosophila melanogaster* images.
3. **Cross-Validation**
    - **Description:** Implement cross-validation to ensure models generalize across different datasets.
    - **Outcome:** Optimized models with improved accuracy and generalization capabilities.
4. **Integration of Analysis Modules**
    - **Description:** Combine different analytical modules (wing segmentation, brain scan classification, fly trajectory analysis) into a cohesive software tool.
    - **Outcome:** An integrated system that provides comprehensive analysis features for researchers.
5. **User Interface Development**
    - **Description:** Develop a web-based interface that allows users to upload images, configure analysis settings, and view results in real-time.
    - **Outcome:** A responsive and user-friendly interface that facilitates easy interaction with the system.
6. **Integration and Testing**
    - **Description:** Integrate all system components and conduct thorough testing, including unit testing, integration testing, and user acceptance testing.
    - **Outcome:** A fully integrated and tested system, ready for deployment.
7. **Deployment and Maintenance**
    - **Description:** Deploy the application on a cloud platform or local servers, provide training to users, and gather feedback for continuous improvement.

- **Outcome:** A live system accessible to users, with ongoing support and maintenance.

Outputs: Reports, Outcomes, Responses, and Notifications

1. Reports

- **Analysis Reports:**
  - **Wing Segmentation Report**
    - **Description:** Provide a report containing labeled segments of healthy or defected wings.
    - **Outcome:** Enables researchers to identify genetic mutations and developmental abnormalities.
  - **Brain Scan Report**
    - **Description:** Present a report detailing the number and types of spots identified in brain scans.
    - **Outcome:** Helps in understanding neurodegenerative diseases and brain functions.
  - **Genotype Classification Report**
    - **Description:** Generate a report predicting the fly's genotype based on flight trajectory analysis.
    - **Outcome:** Supports research on behavioral genetics and phenotype-genotype correlations.
- **Data Export:**
  - **Description:** Allow users to export analysis reports in formats such as PDF or images.
  - **Outcome:** Facilitates sharing and further analysis of research data.

2. Outcomes

- **Automated Analysis Results**
  - **Description:** Provide real-time analysis results based on user inputs and configured parameters.
  - **Outcome:** Reduces manual effort and accelerates research processes.

3. Responses

- **Real-Time Feedback**
  - **Description:** Validate user inputs and provide immediate feedback on errors or missing information.
  - **Outcome:** Reduces data entry errors and ensures accurate and reliable data input.
- **Error Messages and Alerts**

- **Description:** Provide clear and actionable error messages when user actions fail or inputs are invalid.
- **Outcome:** Guides users in correcting mistakes, improving overall system usability.

4. Notifications

- **System Status Updates**
  - **Description:** Inform users about system status changes, such as analysis completion, errors, or downtime.
  - **Outcome:** Keeps users informed and engaged with the system.
- **Model Improvement Notifications**
  - **Description:** Notify users of any updates or improvements made to the models or system.
  - **Outcome:** Encourages users to utilize new features and capabilities.

Non-Functional Requirements (NFRs)

1. **Reliability:**
   - **Input:** Ensure robust data validation to prevent system crashes or data corruption.
   - **Output:** Reports should be generated consistently without failure, providing accurate and reliable insights.
   - **Process:** The system must handle concurrent operations without performance degradation.
2. **Audit Trail:**
   - **Input:** Log user activities related to data entry, updates, and deletions.
   - **Output:** Maintain logs of report generation and access for compliance and security.
   - **Process:** Securely store audit logs to trace user actions and system changes.
3. **Continuity of Processing:**
   - **Input:** Ensure data entry can recover from interruptions without loss.
   - **Output:** Report generation should resume automatically after interruptions.
   - **Process:** Use automatic backups and transaction logging for continuous operations.
4. **Service Level:**
   - **Input:** Ensure 99.9% system availability.
   - **Output:** Deliver reports and notifications within specified timeframes.
   - **Process:** Monitor system performance to meet SLAs.
5. **Methodology:**
   - **Input:** Follow agile development methodologies for continuous integration.
   - **Output:** Ensure quality through automated testing.
   - **Process:** Use DevOps practices for quick deployment and updates.
6. **Correctness:**
   - **Input:** Apply data validation rules to ensure input correctness.
   - **Output:** Reports should accurately reflect the state of analysis.
   - **Process:** Regularly review and test business logic and algorithms.
7. **Ease of Use:**
   - **Input:** Provide an intuitive user interface with feedback mechanisms.
   - **Output:** Simplify report generation with customizable templates.
   - **Process:** Offer user-friendly navigation and help documentation.

8. **Maintainable:**
   - **Input:** Use modular code for easy updates and maintenance.
   - **Output:** Ensure reports and notifications are easily customizable.
   - **Process:** Document the codebase and system components thoroughly.
9. **Portable:**
   - **Input:** Ensure the platform runs on various operating systems.
   - **Output:** Ensure compatibility with different file formats and devices.
   - **Process:** Use platform-independent technologies to support portability.
10. **Coupling:**
   - **Input:** Minimize dependencies between input modules.

- **Output:** Ensure output generation modules are loosely coupled.
- **Process:** Design components to be flexible and scalable.

11. Performance:

- **Input:** Optimize data entry forms for quick responses.
- **Output:** Generate reports and notifications with minimal latency.
- **Process:** Optimize queries and algorithms for performance under load.

12. Ease of Operation:

- **Input:** Provide intuitive data entry fields and options.
- **Output:** Simplify report and notification management tools.
- **Process:** Ensure easy management of operations through a user interface or scripts.

Supervisor Name and signature

MUHAMMAD WASIM