|  |
| --- |
| **Chapter 4**  **System Design** |
| **4.1 Introduction**  Tomato disease detection plays a crucial role in precision agriculture, enabling early identification and intervention to minimize crop losses and ensure food security. Conventional methods for tomato disease detection rely on visual inspection by experts, which is labor-intensive, time-consuming, and prone to human error. To address these limitations, researchers have developed tomato disease detection systems using image processing and machine learning techniques.   * + 1. **Data Collection:** * **Image Acquisition:** Capture high-quality images of tomato leaves under controlled lighting conditions using cameras or mobile devices. * **Image Preprocessing:** Enhance the quality of the acquired images by adjusting brightness, contrast, and color balance, and remove noise. * **Image Segmentation:** Isolate the leaf region from the background, segmenting individual leaves if necessary, to focus the analysis on the areas of interest. * **Feature Extraction:** Extract relevant features from the segmented leaf images, such as texture, color, and shape characteristics, which represent the visual patterns associated with specific tomato diseases.   **4.1.2 Model Training:**   * **Dataset Preparation:** Create a comprehensive dataset of labeled tomato leaf images, representing various diseases and healthy leaves. * **Model Selection:** Choose appropriate machine learning algorithms, such as CNNs, support vector machines (SVMs), based on the nature of the extracted features and the desired accuracy. * **Model Training:** Train the selected machine learning models on the labeled dataset, allowing them to learn the relationship between extracted features and disease labels.   **4.1.3 Disease Detection:**   * **Image Classification:** Apply the trained machine learning models to classify new, unseen tomato leaf images into healthy or diseased categories. * **Disease Identification:** For classified diseased images, identify the specific disease type based on the model's output and associated disease classes. * **Visualization:** Visualizing the detected diseases and providing explanations for the model's decisions.   **4.1.4 Real-world Deployment:**   * **Integration with Agricultural Systems:** Integrate the tomato disease detection system into existing agricultural management systems to facilitate seamless data acquisition, analysis, and decision-making. * **Mobile Application Development:** Develop mobile applications that enable farmers and agricultural workers to capture leaf images and receive real-time disease detection results using their smartphones or tablets. * **Continuous Monitoring and Feedback:** Continuously monitor the performance of the deployed system, collecting feedback from users, and refining the models as needed to maintain high accuracy and generalizability. |
|  |
| * 1. **Architectural Design** |
| C:\Users\DELL\AppData\Local\Packages\Microsoft.Windows.Photos_8wekyb3d8bbwe\TempState\ShareServiceTempFolder\Architecture diagram.drawio.jpeg  **Figure 1: Architecture Diagram**  **4.3 Detailed Design** |
|  |
| **4.3.1 Use Cases**  **4.3.1.1 Farmer**  C:\Users\DELL\AppData\Local\Packages\Microsoft.Windows.Photos_8wekyb3d8bbwe\TempState\ShareServiceTempFolder\Updated Farmer use case.drawio.jpeg  **Figure 1.1: Farmer Use Case**  **4.3.1.2 Admin**  C:\Users\DELL\AppData\Local\Packages\Microsoft.Windows.Photos_8wekyb3d8bbwe\TempState\ShareServiceTempFolder\Admin Use Case 12313.drawio.jpeg  **Figure 1.2: Admin Use Case**   * + 1. **Fully Dressed Use Cases**   **4.3.2.1 Sign Up**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | | Sign Up | | | | **Actors** | | Farmer | | | | **Summary** | | The farmer registers a new account on the platform. | | | | **Pre-Conditions** | | The farmer is not logged in to the platform. | | | | **Post-Conditions** | | The farmer creates an account with basic information like name, email, phone number, and password. | | | | **Special Requirements** | | The email address should be unique and valid. | | | | **Basic Flow** | | | | | | **Actor Action** | | | **System Response** | | | 1 | The farmer opens the Sign Up page. | | 2 | The system displays the Sign Up form. | | 3 | The farmer fills out the form with required information. | | 4 | The system validates the information and creates an account upon successful validation. | | **Alternative Flow** | | | | | | 1 | If the entered email is already registered, the system displays an error message and prompts the farmer to use a different email. | | 2 | If the password doesn't meet the complexity requirements, the system prompts the farmer to choose a stronger password. |  * + - 1. **Login**  |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | | Login | | | | **Actors** | | Farmer, Admin | | | | **Summary** | | The farmer or admin accesses their account by providing login credentials. | | | | **Pre-Conditions** | | The user has a registered account. | | | | **Post-Conditions** | | The user is logged into the platform and has access to their account features. | | | | **Special Requirements** | | The password should be entered correctly. | | | | **Basic Flow** | | | | | | **Actor Action** | | | **System Response** | | | 1 | The user opens the Login page. | | 2 | The system displays the Login form. | | 3 | The user enters their email address and password. | | 4 | The system validates the credentials. | | **Alternative Flow** | | | | | | 1 | If the entered credentials are incorrect, the system displays an error message and allows the farmer to attempt login again. | | 2 | If the farmer forgets the password, there is a "Forgot Password" option that allows them to reset the password |   **4.3.2.3 View Profile**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | | **View Profile** | | | | **Actors** | | Farmer, Admin | | | | **Summary** | | The user views their profile information. | | | | **Pre-Conditions** | | The user having securely authenticated their identity. | | | | **Post-Conditions** | | The user can see their profile details like name, email address, phone number, and other relevant information. | | | | **Special Requirements** | | None | | | | **Basic Flow** | | | | | | **Actor Action** | | | **System Response** | | | 1 | The user clicks on the "Profile" link or icon. | | 2 | The system displays the user's profile page with their information. | | **Alternative Flow** | | | | 1 | None | |   **4.3.2.4 Edit Profile**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | | **Edit Profile** | | | | **Actors** | | Farmer, Admin | | | | **Summary** | | The user updates their profile information. | | | | **Pre-Conditions** | | The user has an access of his/her profile. | | | | **Post-Conditions** | | The user's profile information is updated with the changes. | | | | **Special Requirements** | | None | | | | **Basic Flow** | | | | | | **Actor Action** | | | **System Response** | | | 1 | The user clicks on the "Edit Profile" button. | | 2 | The system displays the editable profile form with the user's current information. | | 3 | The user modifies the desired information on the form. | | 4 | The system validates the input and updates the user's profile with the changes. | | **Alternative Flow** | | | | 1 | If the farmer attempts to submit invalid information, the system displays an error message and prompts the farmer to correct the information. | |  * + - 1. **Upload Images**  |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | | **Upload Images** | | | | **Actors** | | Farmer | | | | **Summary** | | The farmer uploads images of their plants to identify potential diseases. | | | | **Pre-Conditions** | | The farmer accepts the permission to access camera. | | | | **Post-Conditions** | | The system analyzes the images and provides the farmer with potential disease diagnoses and information. | | | | **Special Requirements** | | The image format should be supported by the system. | | | | **Basic Flow** | | | | | | **Actor Action** | | | **System Response** | | | 1 | The farmer clicks on the "Disease Detection" feature. | | 2 | The system displays the image upload interface. | | 3 | \The farmer selects and uploads the images of their plants. | | 4 | The system analyzes the images using image recognition technology. | | **Alternative Flow** | | | | | | 1 | If the uploaded images are of an unsupported format or size, the system prompts the farmer to upload valid images. | | 2 | If there is an issue with the image processing, the system notifies the farmer and recommends re-uploading the images. |  * + - 1. **Access Calendar**  |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | | **Access Calendar** | | | | **Actors** | | Farmer | | | | **Summary** | | The farmer receives a personalized calendar with recommended tasks and schedules for their specific crops based on their location and climate. | | | | **Pre-Conditions** | | The farmer is ready to access the calendar for planning crops and tasks. | | | | **Post-Conditions** | | The farmer has a customized calendar with reminders for essential plant care activities like: **Watering, Fertilization, Pest and disease control, Harvesting** | | | | **Special Requirements** | | Plant care calendar section. | | | | **Basic Flow** | | | | | | **Actor Action** | | | **System Response** | | | 1 | The system automatically generates a personalized calendar after the farmer provides required information. | | 2 | The farmer can filter and prioritize tasks based on urgency or category. | | 3 | The calendar displays tasks with dates, deadlines, and instructions for each activity. | | 4 | The farmer can set reminders for upcoming tasks. | | **Alternative Flow** | | | | 1 | If there are no personalized recommendations for the farmer, the system displays a message suggesting that the farmer has no specific activities scheduled and encourages them to check back later. | |  * + - 1. **Provide Feedback**  |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | | **Provide Feedback** | | | | **Actors** | | Farmer, Admin | | | | **Summary** | | The user provides feedback on the platform's features and functionalities. | | | | **Pre-Conditions** | | The application stands ready to receive and respond to the user's feedback. | | | | **Post-Conditions** | | The feedback is communicated to the admin for review and consideration. | | | | **Special Requirements** | | Feedback form. | | | | **Basic Flow** | | | | | | **Actor Action** | | | **System Response** | | | 1 | The user opens the "Feedback" section. | | 2 | **Describe the issue or suggestion in detail. Attach screenshots or relevant media for clarity.** The system sends the feedback to the admin for review. | | 3 | Submits the feedback. | | 4 | System acknowledges the feedback. | | **Alternative Flow** | | | | 1 | If the feedback form submission fails due to technical issues, the system displays a message informing the farmer about the problem and advises them to try again later. | |  * + - 1. **FAQs**  |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | | **FAQs** | | | | **Actors** | | Farmer, Admin | | | | **Summary** | | The FAQs functionality will allow the user to ask the questions and will get relevant answers about the disease detection. | | | | **Pre-Conditions** | | The User can interact with the FAQs | | | | **Post-Conditions** | | The user receives answers about the asked questions | | | | **Special Requirements** | | Security measures to protect sensitive information. Also Search functionality for users to quickly find relevant FAQs. | | | | **Basic Flow** | | | | | | **Actor Action** | | | **System Response** | | | 1 | User accesses the FAQs page on the website. | | 2 | System displays a list of frequently asked questions. | | **Alternative Flow** | | | | 1 | User cannot find the answer to their question in the FAQs. | |  * + - 1. **View Reports**  |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | | **View Reports** | | | | **Actors** | | Farmer, Admin | | | | **Summary** | | The user accesses and reviews reports generated by the platform to gain insights into their farming practices and crop performance. | | | | **Pre-Conditions** | | Users are all set to open and check out reports in the application. | | | | **Post-Conditions** | | The user gains valuable insights into their farming practices and crop performance based on the generated reports. | | | | **Special Requirements** | | Reports should be generated regularly and updated with new data. | | | | **Basic Flow** | | | | | | **Actor Action** | | | **System Response** | | | 1 | The user accesses the reporting section of the platform. | | 2 | The system displays available reports, categorized by type, such as **Crop Performance Reports, Resource Consumption Reports, Soil Health Reports, Environmental Impact Reports** | | 3 | The user selects a specific report to view. | | 4 | The system displays the report in an interactive format, allowing the user to  **Filter and analyze data based on specific criteria.** | | **Alternative Flow** | | | | | | 1 | The system prompts the user to provide additional information or filter parameters to refine the report. | | 2 | The user provides the requested information or modifies filter settings. | | 3 | The system updates the report with the refined data. | |   **4.3.2.10 Login**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | | **Login** | | | | **Actors** | | Admin | | | | **Summary** | | The administrator accesses the platform's administration panel to manage various functionalities. | | | | **Pre-Conditions** | | The administrator has a registered account and valid login credentials. | | | | **Post-Conditions** | | The administrator is logged in and has access to all administrative features. | | | | **Special Requirements** | | Multi-factor authentication and secure session management. | | | | **Basic Flow** | | | | | | **Actor Action** | | | **System Response** | | | 1 | The administrator opens the Login page. | | 2 | The system displays the Login form. | | 3 | The administrator enters their email address and password. | | 4 | The system validates the credentials. | | 5 | **If credentials are valid** | | 6 | The system logs the administrator into the platform. | | 7 | **If credentials are invalid** | | 8 | The system displays an error message. | | **Alternative Flow** | | | | | | 1 | The system displays an error message and prompts the user to re-enter their credentials. | | 2 | The system informs the user that their account is locked and provides instructions for unlocking it. | | 3 | The system sends a verification code to the administrator's registered phone number or email address for additional security. | |   **4.3.2.11 Manage Diseases**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | | **Manage Diseases** | | | | **Actors** | | Admin | | | | **Summary** | | The administrator adds, edits, and updates information about various diseases affecting crops within the platform. | | | | **Pre-Conditions** | | The admin gains the ability to manage diseases within the application. | | | | **Post-Conditions** | | The administrator manages the disease database, ensuring accurate and up-to-date information for users. | | | | **Special Requirements** | | Integration with a dynamic disease database, and support for classification. | | | | **Basic Flow** | | | | | | **Actor Action** | | | **System Response** | | | 1 | The administrator opens the Disease Management section. | | 2 | The system displays a list of existing diseases. | | 3 | The administrator can:  **Add a new disease, Edit an existing disease, Delete a disease:** | | 4 | The system updates the disease database accordingly. | | **Alternative Flow** | | | | | | 1 | The system prompts the administrator to complete all required fields before submitting the new disease information. | | 2 | The system warns the administrator that the disease already exists and suggests potential actions, such as merging information or creating a new entry with different details. | | 3 | The system validates the format and accuracy of the imported data before adding it to the platform. | |  * + - 1. **Manage Treatments**  |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | | **Manage Treatments:** | | | | **Actors** | | Admin | | | | **Summary** | | The administrator adds, edits, and updates information about various treatment options for different diseases. | | | | **Pre-Conditions** | | The admin is equipped to administer and oversee treatments within the application. | | | | **Post-Conditions** | | The administrator maintains a comprehensive list of treatment options for users to access and utilize. | | | | **Special Requirements** | | Integration with a treatment database, update mechanism, and metrics for effectiveness. | | | | **Basic Flow** | | | | | | **Actor Action** | | | **System Response** | | | 1 | The administrator opens the Treatment Management section. | | 2 | The system displays a list of existing treatment options. | | 3 | The administrator can:  **Add a new treatment, Edit an existing treatment, Delete a treatment** | | 4 | The system updates the treatment database accordingly. | | **Alternative Flow** | | | | | | 1 | The system requires the administrator to provide clear warnings and disclaimers about potential risks associated with the treatment. | | 2 | The system allows the administrator to edit and update the treatment information to ensure accuracy and currency. | | 3 | The system prompts the administrator to provide more reliable references and evidence to support the treatment information. | |   **4.3.2.13 Manage Users**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | | **Manage Users** | | | | **Actors** | | Admin | | | | **Summary** | | The administrator reviews, approves, and manages user accounts within the platform. | | | | **Pre-Conditions** | | The admin is empowered to oversee and manage users within the application. | | | | **Post-Conditions** | | The administrator ensures a safe and productive environment for platform users. | | | | **Special Requirements** | | Role-based access control, activity logs, and customizable profiles. | | | | **Basic Flow** | | | | | | **Actor Action** | | | **System Response** | | | 1 | The administrator opens the User Management section. | | 2 | The system displays a list of all registered users. | | 3 | The administrator can:  **Review user profiles, Approve or reject user registrations, Suspend or ban users, Assign user roles and permissions** | | 4 | The system updates user accounts and access accordingly. | | **Alternative Flow** | | | | | | 1 | The system prompts the user to complete all required information before submitting the registration request. | | 2 | The system may request additional documentation or proof of identity to verify the user's information. | | 3 | The system ensures that user privacy is protected and sensitive information is not publicly accessible. | |   **4.3.2.14 Manage Localized Alerts**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | | **Manage Localized Alerts** | | | | **Actors** | | Admin | | | | **Summary** | | The administrator creates and sends localized alerts to users based on specific criteria. | | | | **Pre-Conditions** | | The admin gains direct authority to manage localized alerts in the application, demonstrating a high level of control and responsiveness. | | | | **Post-Conditions** | | Users receive timely and relevant notifications about potential threats or important information related to their location. | | | | **Special Requirements** | | Geolocation services, weather API integration, and user preferences. | | | | **Basic Flow** | | | | | | **Actor Action** | | | **System Response** | | | 1 | The administrator opens the Localized Alert Management section. | | 2 | The system displays tools for creating and sending alerts. | | 3 | The administrator can:  **Define alert criteria, Create alert messages, Schedule alert delivery.** | | 4 | The system delivers alerts to relevant users via push notifications, email, or other channels. | | **Alternative Flow** | | | | | | 1 | The system informs the administrator that they need more data to define the alert criteria effectively. | | 2 | The system allows users to control their notification preferences and manage the types of alerts they receive. | | 3 | The system alerts the administrator of any technical issues affecting alert delivery and provides troubleshooting steps. | |  * + - 1. **Review Reports**  |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | | **Review Reports** | | | | **Actors** | | Admin | | | | **Summary** | | The administrator analyzes platform usage data and user feedback to gain insights and improve the platform. | | | | **Pre-Conditions** | | The administrator is logged into the platform and has access to reporting dashboards. | | | | **Post-Conditions** | | The administrator identifies trends, patterns, and areas for improvement based on data analysis. | | | | **Special Requirements** | | Comprehensive reporting tools and integration with analytics. | | | | **Basic Flow** | | | | | | **Actor Action** | | | **System Response** | | | 1 | The administrator selects specific reports from a dashboard. | | 2 | The system displays visualizations and data analysis reports on various aspects:  **User activity, Content engagement, Disease management, Chatbot interactions.** | | 3 | The administrator analyzes the data to identify trends, patterns, and areas for improvement. | | 4 | The system logs the administrator's actions and updates the platform accordingly. | | **Alternative Flow** | | | | | | 1 | The administrator investigates further to identify the underlying causes and potential implications. | | 2 | The administrator prioritizes addressing these issues and implementing improvements. | | 3 | The administrator takes immediate action to address the identified risks and protect user data. | |  * + - 1. **Manage FAQs**  |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | | **Manage FAQs** | | | | **Actors** | | Admin | | | | **Summary** | | Admins can log in to the system and manage the FAQs by editing, deleting, or adding new entries. | | | | **Pre-Conditions** | | Admin will manage the FAQs in the system | | | | **Post-Conditions** | | FAQs are successfully managed, and the system reflects the changes. | | | | **Special Requirements** | | None | | | | **Basic Flow** | | | | | | **Actor Action** | | | **System Response** | | | 1 | Admin navigates to the "Manage FAQs" section. | | 2 | System displays a list of existing FAQs with options to edit, delete, or add new entries. | | **Alternative Flow** | | | | | | 1 | None | |   **4.3.3 Database Schema Diagram**  C:\Users\DELL\AppData\Local\Packages\Microsoft.Windows.Photos_8wekyb3d8bbwe\TempState\ShareServiceTempFolder\Database Schema Diagram.drawio.jpeg  **Figure 1.3: Database Schema** |
|  |
| **4.3.4 Methodology Diagram**  C:\Users\DELL\AppData\Local\Packages\Microsoft.Windows.Photos_8wekyb3d8bbwe\TempState\ShareServiceTempFolder\Work Flow.jpeg |

|  |
| --- |
| **Figure 5: Methodology Diagram** |