|  |
| --- |
| **Chapter 5**  **Implementation** |
| **5.1 Endeavour (Team + Work + Way of Working)**  **5.1.1 Tomato Care:**  Nurturing Crops with Innovative Farming Solutions  **5.1.2 Team Members**  Muhammad Salman Afaq (ID: 24775)  Usman Afaq (ID: 24779)  Rafaqat Ahmad (ID: 24784)  **5.1.3 Work Breakdown Structure (WBS)**  **5.1.3.1 Phase 1:** Smart Detection System  **Task 1:** Application Development  **Subtask:** Implementing Real-time Picture Capture  **Subtask:** Developing Disease Detection Model  **Subtask:** Integration with Treatment Recommendations  **Task 2:** Model Training and Dataset Management  Subtask: Training Model with Datasets  **5.1.3.2 Phase 2:** Deployment and Interaction  **Task 3:** Integration and Further Enhancement  **Subtask:** Database Connection  **Subtask:** Google Colab Integration for Model Enhancement  **Subtask:** Continuous Model Improvement  **Task 4:** FAQs  **Subtask:** Developing FAQs for farmer  **5.1.3 Project Schedule**  **5.1.3.1 Milestone 1:** Application Development and Model Training  **Start Date:** [Date]  **End Date:** [Date]  **5.1.3.2 Milestone 2:** Deployment and Optimization  **Start Date:** [Date]  **End Date:** [Date]  **5.1.4 Way of Working**  **5.1.4.1 Agile Methodology**  Overview of Agile Principles Applied  Iterative Development Cycles  Regular Team Reviews and Adjustments  Sprint Durations, Planning Meetings  **5.1.4.2 Collaboration Tools**  **Used Tools:** GitHub for Document and Data Sharing, Google Drive for Collaborative Document Editing  **5.1.5 Risk Management**  **5.1.5.1 Risk 1:** Technical Challenges  **Mitigation Plan:** Regular Team Training Sessions  **Contingency Measures:** External Expertise Consultation’  **5.1.5.2 Risk 2:** Resource Constraints  **Mitigation Plan:** Cross-training Team Members  **Contingency Measures:** Resource Reallocation  **Additional Features:**  **Offline Alerts:** Implementing offline alerts for immediate notifications to farmers.  **Stakeholders:** The primary focus is on farmers, ensuring user-friendly features and effective solutions. |
| **5.2 Flow Control/Pseudo codes**  **5.2.1 Sign up Page**  initialize controllers: username, email, phone, password  displaySignUpPage():  show Screen with Logo, Username, Email, Phone, Password fields, and Sign Up button  onSignUpButtonPress():  if validateInputs(username, email, phone, password):  executeSignUp(username, email, phone, password)  else:  displayErrorMessage("Invalid inputs")  validateInputs(username, email, phone, password):  return true if inputs are valid, else false  executeSignUp(username, email, phone, password):  createNewUser locally  sendSignUpRequestToServer  navigateToHomePage  displayErrorMessage(message):  show Error Message  createNewUser(username, email, phone, password):  // Local user creation logic  sendSignUpRequestToServer(username, email, phone, password):  // Send user details to the server  navigateToHomePage():  show Home Page  **5.2.2 Login Page**  # Initialize user input controllers  initialize controllers: username, password  # Display the login page  displayLoginPage():  show Screen with:  - Logo  - Username input field  - Password input field (masked)  - Login button  # Handle Login button press  onLoginButtonPress():  if validateInputs(username, password):  executeLogin(username, password)  else:  displayErrorMessage("Invalid credentials. Please check and try again.")  # Validate user inputs  validateInputs(username, password):  if any input is empty or invalid:  return false  else:  return true  # Execute login logic  executeLogin(username, password):  if authenticateUser(username, password):  navigateToHomePage()  else:  displayErrorMessage("Authentication failed. Please try again.")  # Authenticate user  authenticateUser(username, password):  # Logic to check username and password against stored credentials or server  # Display error message  displayErrorMessage(message):  show Error Message on Screen  # Navigate to the home page after successful login  navigateToHomePage():  show Home Page |
| **5.3 Components, Libraries, Web Services, and Stubs**  **5.3.1 Components**  **5.3.1.1 CNN Model**  Description: Core for disease detection.  Functionality: Implements CNN for image classification.  Implementation: TensorFlow and Python.    **5.3.1.2 Flask Web Application**  Description: Web interface for CNN interaction.  Functionality: Processes image data.  Implementation: Flask in Python.  **5.3.1.3 Flutter Application**  Description: Mobile app for real-time monitoring.  Functionality: Integrates with Flask for interaction.  Implementation: Flutter SDK.  **5.3.2 Libraries**  **5.3.2.1 TensorFlow**  Description: Deep learning library.  Integration: Constructs and trains CNN.  **5.3.2.2 OpenCV**  Description: Image processing library.  Integration: Used for image pre-processing.  **5.3.2.3 Flutter SDK**  Description: Cross-platform app development kit.  Integration: Utilized for Flutter app.  **5.3.3 Web Services**  **5.3.3.1 Flask API**  Description: Interface for CNN-Flutter communication.  Implementation: Created using Flask.  **5.3.3.2 Fire Base Database**  Description: Stores disease data.  Integration: Utilizes FireBase Database  **5.3.4 Stubs**  **5.3.4.1 Testing Stubs**  Description: Simulates external components for testing.  Usage: Validates component functionality.  **5.4 IDE, Tools and Technologies**  **5.4.1** Optimized Technology Stack for Tomato Care Application  **5.4.2 Integrated Development Environment (IDE)**  Backend: Python (Specifically for Backend Development)  Flutter: Visual Studio Code  Collaborative Model Training: Google Colab  **5.4.2.1 Tools:**  Version Control and Document Sharing: GitHub  Containerization: Not Applicable (Omitted Docker for simplicity)  **5.4.2.2 Technologies:**  Backend: Python (Specifically for Backend Development)  Framework: Flask (Backend Web Framework)  Machine Learning: TensorFlow, OpenCV  Database: Firebase  Flutter: Dart  **5.5 Best Practices and Coding Standards**  **5.5.1 Software Engineering Practices**  Version Control: GitHub  Continuous Integration and Deployment Practices  **5.5.2 Development Practices and Standards**  Python Coding Standards: Adherence to PEP 8  Flutter: Followed Flutter's Coding Standards  **5.6 Deployment Environment for Tomato Care Application**  **5.6.1 Deployment Environment:**  **5.6.1.1 Operating System:** Not Specified (Omitted AWS EC2  instances for simplicity)  **5.6.1.2 Server Configuration:**  Not Specified (Omitted specific server configuration for simplicity)  **5.6.1.3 Database:**  Firebase Database |
| **5.8 Summary**  In this chapter, we outlined the composition of our project team, the detailed work breakdown structure for the implementation phase, and the methodologies and tools adopted for efficient collaboration. We addressed potential risks and mitigation strategies, presented flow control or pseudo codes, listed key components and technologies, and detailed the deployment environment. Our team adhered to best practices and coding standards throughout the implementation, ensuring a robust and effective solution.  **5.8.1 References**  **[1]** Durmuş, H., Güneş, E. O., & Kırcı, M. (2017, August). Disease detection on the leaves of the tomato plants by using deep learning. In 2017 6th International conference on agro-geoinformatics (pp. 1-5)  **[2]** Ashqar, B. A., & Abu-Naser, S. S. (2018). Image-based tomato leaves diseases detection using deep learning.  **[3]** Tm, P., Pranathi, A., SaiAshritha, K., Chittaragi, N. B., & Koolagudi, S. G. (2018, August). Tomato leaf disease detection using convolutional neural networks. In 2018 eleventh international conference on contemporary computing (IC3) (pp. 1-5)  **[4]** Chowdhury, M. E., Rahman, T., Khandakar, A., Ibtehaz, N., Khan, A. U., Khan, M. S., ... & Ali, S. H. M. (2021). Tomato leaf diseases detection using deep learning technique. Technology in Agriculture, 453.  **[5]** Harakannanavar, S. S., Rudagi, J. M., Puranikmath, V. I., Siddiqua, A., & Pramodhini, R. (2022). Plant leaf disease detection using computer vision and machine learning algorithms. Global Transitions Proceedings, 305-310.  **[6]** Gadade, H. D., & Kirange, D. K. (2021, April). Machine learning based identification of tomato leaf diseases at various stages of development. In 2021 5th International Conference on Computing Methodologies and Communication (ICCMC) (pp. 814-819). IEEE. |