Intel Unnati Industrial Training Program 2024

"Customized AI Kitchen for India"

BACHELOR OF ENGINEERING

In

ELECTRONICS AND COMMUNICATIONS ENGINEERING

By

ABHISHEK DHONDIRAM SHINDE

1EP22EC001

4th sem ECE

Under the guidance of

Parvati Patil
Department of ECE, EPCET.

EAST POINT COLLEGE OF ENGINEERING & TECHNOLOGY

Jnana Prabha, East Point Campus, Virgonagar Post, Avalahalli, Bangalore-560049
Department of Electronics & Communication Engineering 2023-2024

ACKNOWLEDGEMENT

Any achievement, be it scholastic or otherwise doesn't depend solely on the individual efforts but on the guidance, encouragement and cooperation of the intellectuals, elders and friends. I would like to take this opportunity to thank them all.

I express my gratitude to **Dr. Mrityunjaya V Latt**e, Principal, EPCET for his kind words of support and encouragement.

I express our sincere regards and thanks to **Dr. Yogesh G S**, Vice Principal and Head of the department, ECE, EPCET.

I'm grateful to acknowledge the guidance and encouragement that has been given to me by, **Parvati Patil ,** Assistant Professor ,ECE, EPCET, who has rendered valuable assistance.

I'm obliged to **Abhishek Nandy,** our External Project Mentor , who have helped us in several ways to learn and explore things by guiding us with all the required support.

I extend my sincere thanks to the department of ECE, EPCET, who have encouraged us throughout the course. I also express my deep sense of obligation to my parents and God for their consistent blessings and encouragement.

INTRODUCTION

The introduction to customised AI kitchen simulation using LabVIEW sets the stage for the project and its objectives:

“REVOLUTIONIZING COOKING WITH INTELLIGENT AUTOMATION”

Designing a customized AI-powered kitchen for India involves considering the unique culinary practices, ingredient availability, and cooking styles prevalent in Indian households. Here are some features that could be included:

1.**Smart Recipe Recommendations**

- Personalized Recipes: Based on dietary preferences, health goals, and local cuisine.
- Ingredient Substitution: Suggest alternatives based on what's available in the kitchen.

2. Voice-Activated Assistant

- Multilingual Support: Capable of understanding and responding in multiple Indian languages.
- Step-by-Step Cooking Guidance: Real-time instructions and tips for traditional and modern recipes.

3. Inventory Management

- Smart Pantry: Automatically tracks stock levels of ingredients and suggests shopping lists.
- Expiration Alerts: Notifies users of soon-to-expire items to reduce food waste.

4. Smart Appliances Integration

- IoT-Enabled Appliances: Ovens, stoves, and refrigerators that can be controlled via a central AI.
- System.Energy Efficiency: Optimizes energy use based on cooking habits and times.

5.Nutritional Insights

- Calorie and Nutrition Tracking: Provides information about the nutritional content of meals.
- Dietary Suggestions: Offers meal plans based on health conditions like diabetes or hypertension.

6. Cultural and Regional Adaptations

- Traditional Recipes Database: A vast collection of regional Indian recipes, including festival and seasonal dishes.
- Cooking Techniques: Guides on traditional methods like tandoori, tadka, and more.

Implementing these features requires a combination of advanced AI algorithms, robust hardware, and seamless integration with various smart devices and services.

PROBLEM STATEMENT: CHALLENGES IN INDIAN KITCHENS AND CUSTOMISING THE KITCHEN

CHALLENGES IN INDIAN KITCHEN

1. Diverse Culinary Practices

- Regional Variations: India's cuisine varies significantly across regions, with different ingredients, cooking methods, and flavours.
- Dietary Preferences: There are numerous dietary preferences including vegetarianism, veganism, and various religious dietary restrictions.

2. Ingredient Availability

- Local and Seasonal Ingredients: Availability of ingredients can be seasonal and region-specific.
- Substitutions: Finding suitable substitutes for traditional ingredients that may not be available everywhere.

3. Cooking Methods

- Traditional Techniques: Many traditional Indian cooking methods (e.g., tandoor, pressure cooking, tadka) require specific equipment and skills.
- High-Heat Cooking: Indian cooking often involves high-heat techniques that need specialized appliances.

CUSTOMISING THE KITCHEN

1. AI-Driven Recipe Recommendations

- Regional Recipes: Include a vast database of region-specific recipes and cooking techniques.
- Dietary Customization: Allow users to filter recipes based on dietary preferences and restrictions.

2. Smart Ingredient Management

- Local Substitutes: AI can suggest local substitutes for ingredients that are not available.
- Seasonal Adjustment: Recommend recipes based on seasonal availability of ingredients.

3. Advanced Cooking Appliances

- Multi-functional Devices: Equip kitchens with multi-functional appliances that save space and energy.
- High-Heat Cooking Solutions: Include specialized appliances for high-heat cooking methods like tandoors and woks.

TECHNOLOGY USED: Labview SIMULATOR

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) is a graphical programming environment used for data acquisition, instrument control, and industrial automation. It is known for its visual programming language that allows users to build programs by connecting functional blocks, representing the flow of data.

Key Components

1. Front Panel:

- The user interface of a LabVIEW program.
- Contains controls (inputs) like knobs, buttons, and sliders, and indicators (outputs) like graphs, LEDs, and numeric displays.

2.Block Diagram:

- The back-end where the program logic is created.
- Consists of graphical code including structures, functions, and wires that represent data flow.

Fundamental Concepts

Graphical Programming:

- Uses visual elements to create programs, making it accessible for those unfamiliar with traditional text-based programming languages.

Data Flow:

- Execution of code is determined by the flow of data through the nodes. A node executes when all its input data is available, and it generates output data for subsequent nodes.

Virtual Instruments (VIs):

- A program in LabVIEW is called a VI, consisting of a front panel and a block diagram.

Basic Elements

Controls and Indicators:

- Controls: Inputs to the program (e.g., buttons, numeric inputs).
- Indicator: Outputs from the program (e.g., graphs, numeric displays).

Structures:

- While Loop: Repeats the code inside it until a specified condition is met.
- For Loop: Repeats the code a set number of times.
- Case Structure: Executes different code based on the value of an input.

Functions:

- Pre-built blocks for performing specific operations such as mathematical computations, data acquisition, and file I/O.

Simulation in LabVIEW

Simulated I/O:

- LabVIEW allows users to simulate input and output signals, facilitating testing and debugging without the need for actual hardware.
- The Simulate Signal Express VI can generate waveforms and other test signals.

Example: Simple Addition VI

1. Front Panel:

- Add two numeric controls for inputs.
- Add a numeric indicator for the result.

2.Block Diagram:

- Place an addition function.
- Wire the two numeric controls to the inputs of the addition function.
- Wire the output of the addition function to the numeric indicator.

3. Run the VI:

- Input values in the numeric controls and observe the result in the numeric indicator.

Debugging Tools

- Probes: Monitor data values as they flow through the wires.
- Breakpoints: Pause execution at specific points to examine the state of the VI.
- Execution Highlighting: Visually follow the data flow and see how the data moves through the VI.

LABVIEW INTEGRATION: "Leveraging LabVIEW for AI Kitchen Development"

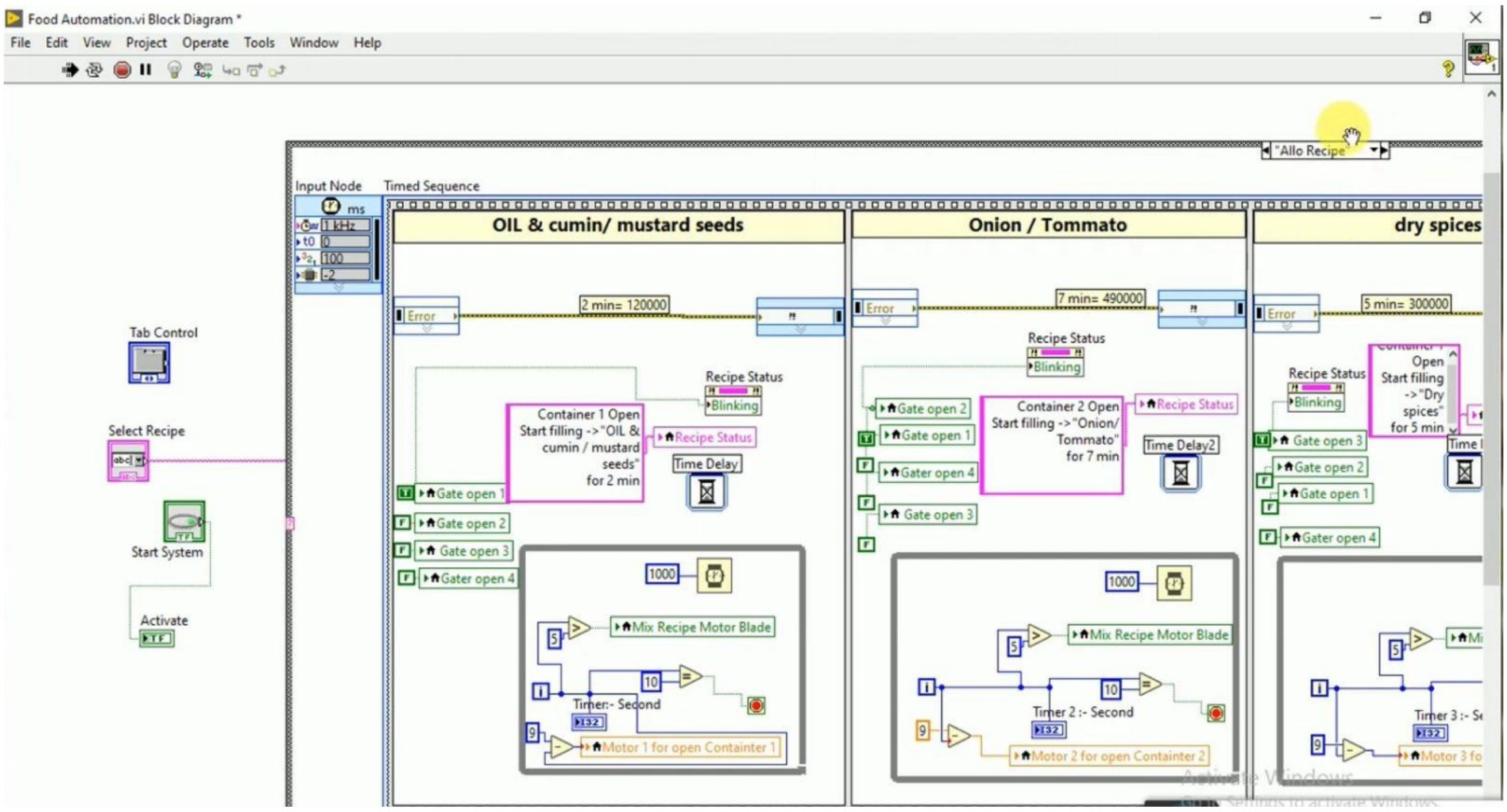

Figure 1: Block Diagram Panel of CUSTOMIZATION OF KITCHEN USING AI

We can understand the block diagram of the system by breaking it down in terms of its functionality:

To integrate the LabVIEW program step by step based on the given block diagram:

Initial Setup

1. Input Node:
   - Configure input parameters (e.g., time in ms, kHz, etc.).
   - Use "Tab Control" to select different recipes or sequences.

2. Select Recipe:
   - Use a dropdown menu to select the desired recipe.
   - Connect this to the "Start System" block.

3. Start System:
   - Activate the system to begin the sequence.

Oil & Cumin/Mustard Seeds Handling

1. Container 1 Open:
   - Set the timer to 2 minutes (120000 ms).
   - Open Container 1 to start filling oil and cumin/mustard seeds.
   - Update "Recipe Status" to blinking.
   - Manage gates for ingredient flow (Gate 1, 2, 3, 4).

2. Mix Recipe Motor Blad:
- Start the motor blade for mixing.
- Control Motor 1 for opening Container 1.

Onion/Tomato Handling

1.Container 2 Open:
- Set the timer to 7 minutes (490000 ms).
- Open Container 2 to start filling onions/tomatoes.
- Update "Recipe Status" to blinking.
- Manage gates for ingredient flow (Gate 2, 3, 4).

2. Mix Recipe Motor Blade:
- Start the motor blade for mixing.
- Control Motor 2 for opening Container 2.

Dry Spices Handling

1. Container 1 Open:
- Set the timer to 5 minutes (300000 ms).
- Open Container 1 to start filling dry spices.
- Update "Recipe Status" to blinking.
- Manage gates for ingredient flow (Gate 3, 1, 4).

2. Mix Recipe Motor Blade:
- Start the motor blade for mixing.
- Control Motor 3 for opening Container 3.

Sliced Vegetables & Water Handling
- Container 1 Open:
- Set the timer to 2 minutes (120000 ms).
- Control motor 4 to open container 1.
- Update Recipe Status to "Blinking".
- Open gates for vegetables and water.

Sequence Control and Timers

1. Timer:
   - Set timers for each process step (e.g., Timer 1 for oil, Timer 2 for onions, Timer 3 for spices).

2. Error Handling:
   - Include error indicators and handlers to ensure smooth operation.

3. Timed Sequence:
   - Use timed sequences to ensure each step executes in the correct order.

Final Integration

1.Connect All Blocks:

- Ensure all blocks are properly connected to manage the flow of the process.
- Update status indicators to reflect the current process stage.

2. Testing and Debugging:

- Run the program to identify and fix any issues.
- Ensure all steps work seamlessly together.

This approach ensures a systematic and efficient integration of the LabVIEW program for automating kitchen processes.

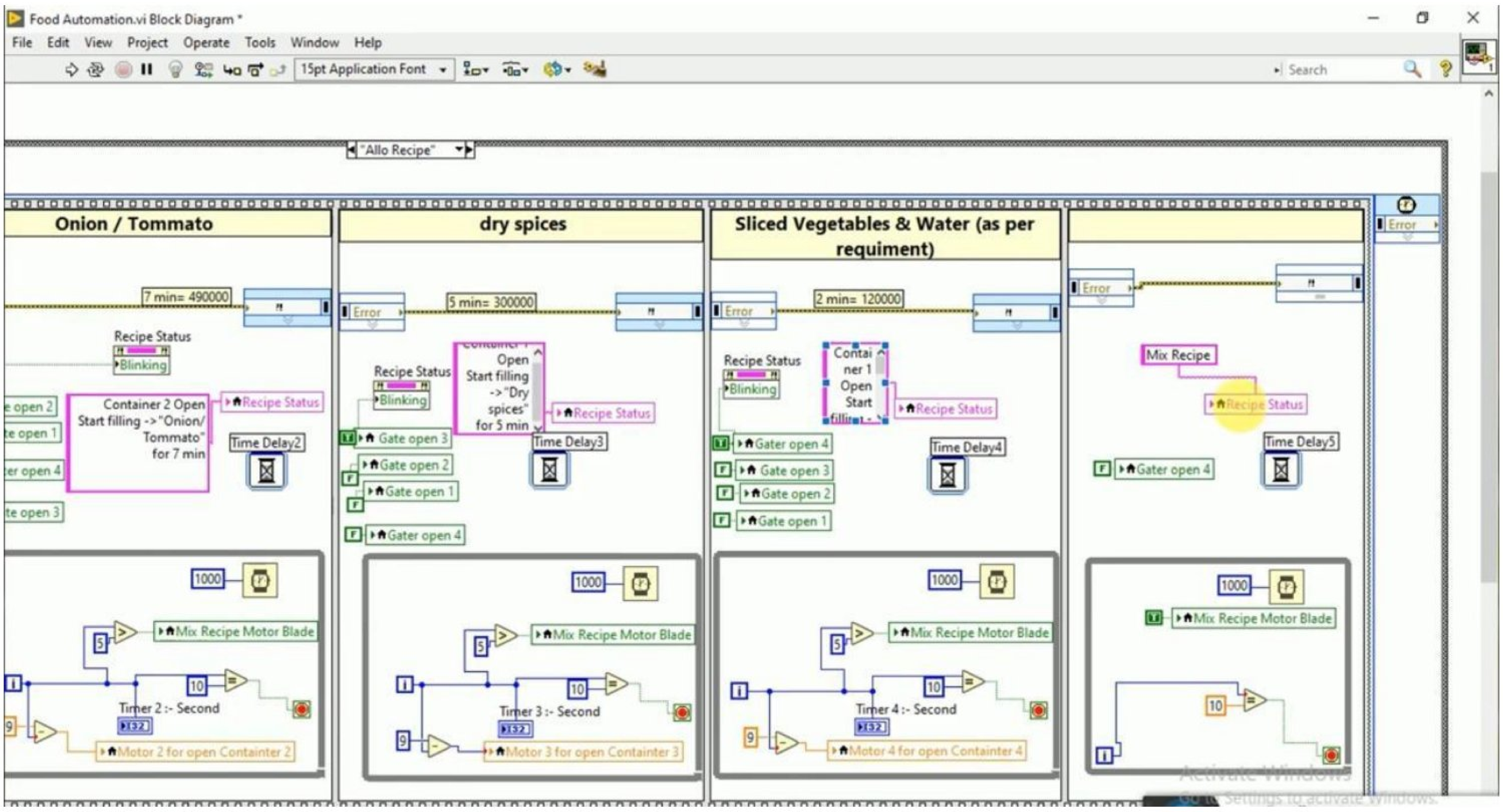

Figure 2: Block Diagram Panel of CUSTOMIZATION OF KITCHEN USING AI

1. Timed Sequence and Input Nod:

- The input node on the left is set up to control the timing of the process, indicating that the sequence runs every 100 milliseconds.
- The Timed Sequence structure is used to control the execution of the recipe steps in a deterministic manner.

2. Recipe Selection:

- There is a control for selecting a recipe (Rice Recipe) with different options (False, Alio Recipe, Rice Recipe, Empty).

3. Process Steps:

- The diagram is divided into two main sequences:
- Sliced Vegetables & Water:
- This sequence controls the opening of container gates (Gate open 1, Gate open 3, Gate open 4) to fill a container with water or sliced vegetables based on the recipe.
- A container (Container 4) is opened to start filling with Rice / Water for 5 minutes.
- Time Delay blocks are used to introduce delays in the sequence.

2. Mix Recipe:

- This sequence involves mixing the ingredients.
- A Mix Recipe Motor Blade is activated, followed by the operation of a motor to open Container 4.

4. Error Handling:

- Error wires are used throughout the diagram to handle and propagate errors during the execution.

5. Recipe Status:

- The status of the recipe execution is monitored and updated.

This LabVIEW application seems to automate the process of preparing a food item based on selected recipes by controlling various actuators (gates, motors) and using a timed sequence structure to ensure proper timing and synchronization of actions.

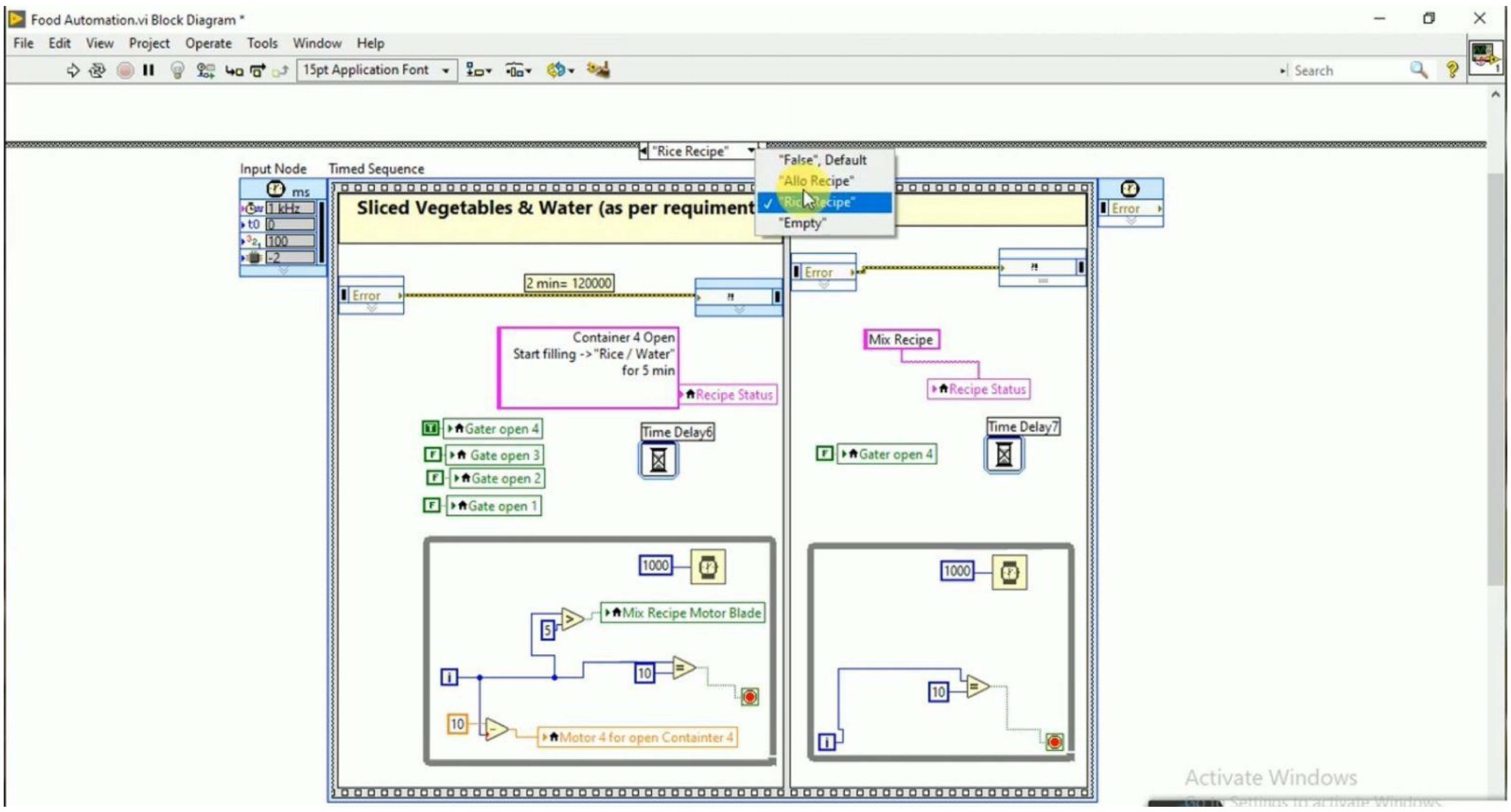

Figure 3: Block Diagram Panel of CUSTOMIZATION OF KITCHEN USING AI

FRONT PANEL: Working Of The Simulation

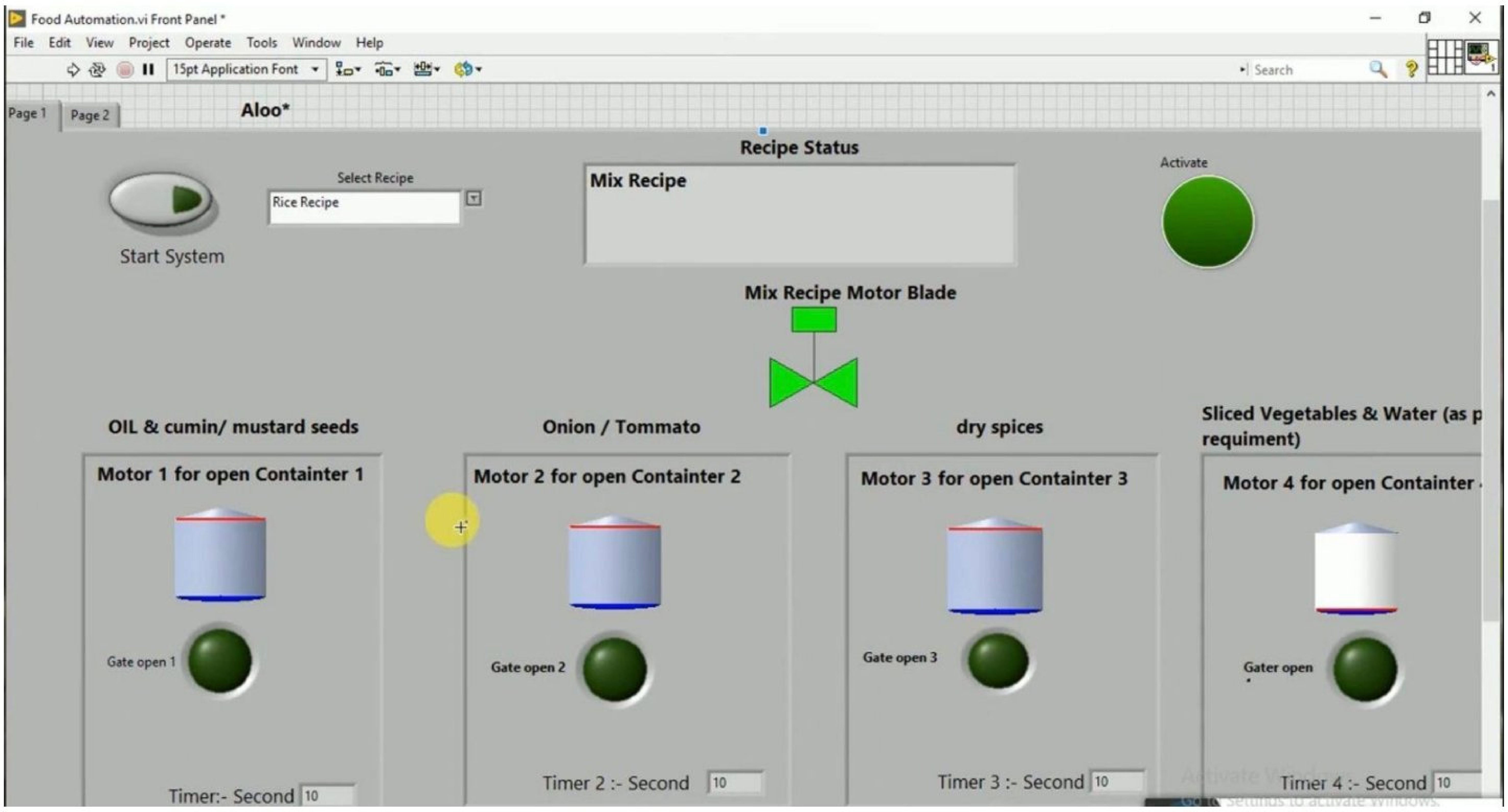

Figure 4: Front Diagram Panel of CUSTOMIZATION OF KITCHEN USING AI

The figure illustrates the front panel of a LabVIEW program, which appears to be designed for automating a food preparation process. Here is a step-by-step interpretation of the interface:

Start System:

A button to initiate the food preparation process.
Select Recipe:

A dropdown menu to select the recipe, currently set to "Rice Recipe".
Recipe Status:

A display area to show the current status of the recipe, which is "Mix Recipe".
Activate:

A button or indicator to activate or show that the system is active.
Mix Recipe Motor Blade:

An indicator showing the status of the motor blade used to mix the recipe.
Containers and Motors:

Four containers labeled for different ingredients:

Container 1: For "OIL & cumin/mustard seeds" with "Motor 1 for open Container 1" and "Gate open 1".
Container 2: For "Onion / Tomato" with "Motor 2 for open Container 2" and "Gate open 2".
Container 3: For "dry spices" with "Motor 3 for open Container 3" and "Gate open 3".
Container 4: For "Sliced Vegetables & Water (as per requirement)" with "Motor 4 for open Container 4" and "Gate open".
Timers:

Each container has an associated timer to control how long the gate remains open:
Timer 1: For Container 1, set to 10 seconds.
Timer 2: For Container 2, set to 10 seconds.
Timer 3: For Container 3, set to 10 seconds.
Timer 4: For Container 4, set to 10 seconds.
This interface allows the user to control and monitor the automated process of adding ingredients and mixing them according to the selected recipe. Each container is equipped with a motor to open it and a timer to control the duration of the operation. The overall system status and specific actions (like mixing) are displayed for easy monitoring.

TEAM MEMBERS AND CONTRIBUTION

LabVIEW simulation of the project: Abhishek Dhondiram Shinde

Project report: Abhishek Dhondiram Shinde

CONCLUSION:

LabVIEW simplifies the process of creating, simulating, and debugging programs through its graphical interface and data flow programming model. It is particularly useful in engineering and scientific applications where data acquisition and instrument control are essential.

Creating a conclusion for a project on customizing an AI kitchen using a LabVIEW simulator involves summarizing key points, achievements, and potential future work. Here is a sample conclusion:

The customization of an AI-powered kitchen using the LabVIEW simulator has been a comprehensive and innovative project. This undertaking demonstrates the potential of integrating advanced AI technologies with modern kitchen appliances to enhance functionality, efficiency, and user experience.

Key Achievements:

1. Integration and Automation: Successfully integrated various kitchen appliances with AI capabilities, allowing for automated control and monitoring through the LabVIEW simulator. This setup ensures that tasks such as cooking, cleaning, and inventory management are performed with minimal human intervention.

2. User-Friendly Interface: Developed a user-friendly interface that allows users to interact with the AI kitchen easily. The interface provides real-time feedback and control options, enhancing the overall user experience.

3. Data-Driven Insights: Leveraged data collection and analysis to optimize kitchen operations. The AI system can predict maintenance needs, suggest recipes based on available ingredients, and improve energy efficiency by learning usage patterns.

4. Safety and Security: Implemented robust safety and security features. The AI kitchen can detect potential hazards, such as gas leaks or fire, and take appropriate actions to mitigate risks.

Challenges and Solutions:

Integration Complexity: The integration of diverse appliances and technologies posed significant challenges. These were addressed by developing modular interfaces and using standardized communication protocols.

User Adaptation: Ensuring that users can adapt to the AI kitchen required extensive usability testing and iterative design improvements based on user feedback.

System Reliability: Maintaining system reliability and robustness was achieved through rigorous testing and the implementation of redundant systems to handle failures gracefully.

Future Work:

1. Enhanced AI Capabilities: Future enhancements could include more advanced AI algorithms for better decision-making and predictive analytics, further improving efficiency and user experience.

2. Expanded Appliance Integration: Integrating additional kitchen appliances and expanding the system's capabilities to cover more complex tasks and workflows.

3.Scalability and Customization: Developing scalable solutions that can be customized to different kitchen sizes and user needs, making the AI kitchen accessible to a broader audience.

4. Sustainability: Incorporating more sustainable practices and technologies, such as optimizing energy usage and reducing waste, to align with global sustainability goals.

In conclusion, the AI kitchen customized using the LabVIEW simulator represents a significant step forward in the evolution of smart home technology. It not only enhances the convenience and safety of kitchen operations but also sets the stage for future innovations in home automation. Continued development and refinement of this technology will likely lead to even more sophisticated and efficient kitchen environments, ultimately transforming how we interact with our home spaces.