**Task 1: VIP Food App Integration**

To integrate the fictitious VIP Food App into the access management system, create an application registration in Azure AD. This will enable role-based access to app functionalities based on the logged-in user’s assigned role.

- Application Registration:

* Register the VIP Food App in Azure AD.
* Configure API permissions to allow access to specific app features.
* Use role-based claims to ensure users receive access only to functionalities relevant to their roles (e.g., chefs accessing kitchen management tools, catering managers handling logistics).

- Role-Based Access:

* Assign roles such as ChefRole, CateringManagerRole, and OfficeRole within the app.
* Ensure seamless integration by mapping Azure AD roles to app-specific permissions.

**Task 2: Custom Azure Roles**

Define custom Azure roles that align with the distinct responsibilities of each user group at VIP Events. These roles should adhere to the principle of least privilege, ensuring users have only the permissions necessary for their job-related activities.

1. Custom Role Design:

- Create roles such as:

* EquipHandlersRole: For equipment handlers managing dock operations and storage.
* ChefsRole: For chefs preparing food and managing kitchen systems.
* OfficeRole: For office workers handling administrative tasks.
* ManagementRole: For the CEO and senior management overseeing strategic operations.
* TransientStaffRole: For temporary staff performing day-specific tasks.

- Role Alignment:

* Align roles with the subnets and wireless segments defined in Stage 2 (e.g., Dock\_Operations, Kitchen, Office).
* Ensure roles are scoped to specific resources (e.g., IoT devices, kitchen management tools).

**Task 3: Assigning Permissions to Azure Roles**

Explicitly define the permissions granted to each custom Azure role. Balance security and operational efficiency by carefully selecting permissions.

1. Permissions for Each Role:

- EquipHandlersRole:

* Read/write access to dock operations and equipment storage systems.
* No access to sensitive financial or administrative data.

- ChefsRole:

* Access to kitchen management tools and inventory tracking systems.
* Permission to view and update recipes but not modify financial records.

- OfficeRole:

* Access to administrative systems, file storage, and email services.
* Limited access to operational systems like dock or kitchen tools.

- ManagementRole:

* Full access to all systems, including financial and operational data.
* Ability to manage Azure AD configurations and assign roles.

- TransientStaffRole:

* Temporary access to guest Wi-Fi and limited event management tools.
* Permissions expire automatically after a predefined period.

2. Security Measures:

- Enforce multi-factor authentication (MFA) for all roles.

- Use conditional access policies to restrict access based on location, device compliance, and risk levels.

**Task 4: Mapping User Groups to Azure Roles**

Systematically assign Azure roles to the user groups defined in Stage 2. This ensures each employee group has access permissions tailored to their roles and responsibilities.

1. Mapping:

- Equipment Handlers: EquipHandlersRole

- Chefs and Head Chef: ChefsRole

- Office Workers: OfficeRole

- CEO: ManagementRole

- Transient Staff: TransientStaffRole

2. Implementation:

- Use Azure AD group-based role assignments to streamline access management.

- Automate role assignments for transient staff using dynamic groups and time-limited permissions.

**Documentation Summary**

Extended proposal with comprehensive documentation for Azure role configuration. Including:

- Custom Azure Roles:

* Role names, descriptions, and associated permissions.

- Permission Details:

* Explicit list of permissions for each role, adhering to the principle of least privilege.

- User Group Mappings:

* Clear mapping of user groups to Azure roles, ensuring alignment with job responsibilities.

- Integration Details:

* Application registration for the VIP Food App and role-based access within the app.