**Project Concept Memo**

**Project Topic: Simple Contact Management System**

The project I propose is a "Simple Contact Management System." This system will allow users to manage their personal and professional contacts efficiently. The primary goal of the system is to provide an easy-to-use interface for users to add, update, and organize their contact information.

**Data and Relationships**

The system will involve the following main entities:

**Users:** This table will store information about the users, including UserId, UserName, and Password.

**Contacts:** This table will store information about the contacts, including ContactId, UserId, ContactName, PhoneNumber, Email, and Address.

**Groups**: This table will allow users to categorise their contacts, including GroupId, UserId, GroupName, Description, MaxContacts, and CreatedAt.

**ContactGroups**: This table will manage the many-to-many relationship between Contacts and Groups, including ContactGroupId, ContactId, and GroupId.

**ContactNotes:** This table will allow users to add notes to their contacts, including NoteId, ContactId, NoteText, and CreatedAt.

**The relationships between these entities are as follows:**

* Each user can have multiple contacts.
* Each user can have multiple groups.
* Each contact can belong to multiple groups, and each group can contain multiple contacts (many-to-many relationship).
* Each contact can have multiple notes.

The UserId field in the Contacts and Groups tables will establish the relationship, linking contacts and groups to their respective users. The ContactId and GroupId fields in the ContactGroups table will establish the many-to-many relationship between Contacts and Groups. The ContactId field in the ContactNotes table will link notes to their respective contacts.

**Business Rules**

To ensure data integrity and enforce business logic, the following business rules will be implemented:

**Duplicate Contact Rule:** Prevent the addition of duplicate contacts for the same user based on a combination of key fields.

**Description:** To avoid redundancy and confusion, a user should not be able to add a contact that is essentially the same as an existing one. This rule will check for duplicates based on the combination of ContactName, PhoneNumber, and Email.

Enforcement: When a new contact is being added or an existing contact is being updated, the system will search for existing contacts for the same user that have the same ContactName, PhoneNumber, and Email. If a match is found, an error message will be displayed, and the addition or update will be prevented.

**Maximum Contacts per Group Rule:** Ensure that no group exceeds its predefined maximum number of contacts.

**Description:** To maintain manageability and usability, each group will have a maximum number of contacts defined by the MaxContacts field. This rule ensures that groups remain organized and not overloaded with contacts.

Enforcement: When a contact is being added to a group, the system will check the current number of contacts in that group against the MaxContacts value. If adding the new contact would exceed the maximum allowed, an error message will be displayed, and the addition will be prevented.

**Conclusion**

This project will leverage a structured database to manage contacts efficiently while enforcing robust business rules to maintain data integrity and support business logic. By implementing these rules, we ensure that the system remains reliable, user-friendly, and capable of supporting users in organizing their contact information effectively.