Bumble Simulation
Database Design Document (DDD)

Version 1.0

Prepared by: Reyes, Lovely Joy P.

Revision History

Date	Version	Description	Author
2/26/2025	1.0	Made a polish and correction to some logics and conditions to methods and statements.	Lovely P. Reyes

Table of Contents

1 Introduction

The section introduces the Database Design Document (DDD) for Bumble Simulation.

"Make the first move." Is the most known slogan of the known dating application in the world, Bumble. The intention of this application is to assist you in relating to various individuals according to your own preferences. It's similar to Do It Yourself (DIY), but it's employed for networking, forming friendships, and even for recreational causes. Globally, it has more than 3 million users. The intriguing aspect of it is that you might occasionally run into individuals you know well.

Making a mocked-up Bumble application is just as fascinating as it seems. My objective is to build a database from scratch and make the application similar to Bumble using all the resources we are permitted to utilize. The Bumble Simulation Database Design Document (DDD) enables the display of the data being used and the database that was created for the application. Additionally, it will draw attention to the useful logic flow for tables across the program. Finally, it will refine the logical mapping created to run the application with proper access, manipulation, and implementation.

1.1 Document Objectives

The objective of this document is to:

- To function as a reference on how appropriate use impacts the database and data material specifications.

- Assess the significance of using a database to store data and information, particularly for the security of applications and personal data.

- Determine how we can access and connect the created and implemented tables and database.

- Design an Entity Relationship Diagram (ERD) that makes it simpler to convey the database table's structure.

1.2 Intended Audiences

This DDD is intended for the following audiences:

- **Subject professor/Lecturer** who provided the project and will be examining the application's general features and functionality.
- **Co-learners**, who will be watching the defense and carry out the same assignment and offer criticism.
- **Users**, who is going to use the simulated application that this paper covers.

1.3 References

This DDD refers to the following references:

- Bumble Inc.
- Bumble Phone and Desktop Application

2 Detailed Database Design

This part will show the detailed design of different database table with their connections and relationships.

2.1.1 Data dictionary

2.1.1.1 Data dictionary for Element: bumbleusers

Name	Data Type	Constrain	Description
User_id (primary key)	Integer	Min :1, Max:1	Users' unique identification in the simulation.
Fname	Varchar		Chosen name of the user.
Birthday	Date	Min :1, Max:1	Birthday of the user with the format YYYY-MM-DD.
Age	Integer		Age of the user.
Password_hash	Varchar		Stores the user's chosen password which is being hashed and should be 8 characters long.
Email	Varchar		User personal email.

2.1.1.2 Data dictionary for Element: userpreference

Name	Data Type	Constrain	Description
Matches_ID (primary key)	Integer	Min :1, Max:1	Unique identification for users matches.
User_ID (foreign key)	Integer		This column is being referred to the table bumbleusers. This is the account used to log in.
Matched_user_ID	Integer		This is the other stored user in the database that matched with the user's.
Match_date	Timestamp		Store the date matches happened.

2.1.1.3 Data dictionary for Element: usermessages

Name	Data Type	Constrain	Description
Message_ID (primary key)	Integer		Unique identification for the message transaction.
matches_ID (foreign key)	Integer	Min :1, Max:1	Adopted from the userpreference table.
Sender_ID (foreign key)	Integer		The sender of the message.
Message_text	Text		The message sent.
Message_date	Timestamp		Time the message was sent.

2.1.1.4 Data dictionary for Element: subsfeat

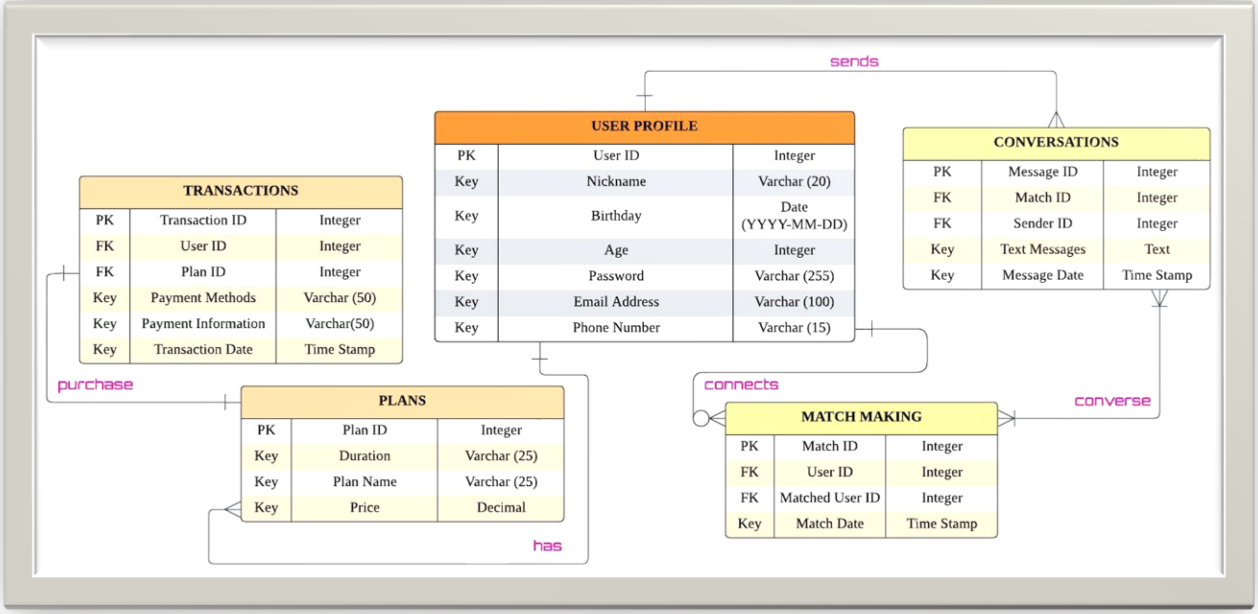
Name	Data Type	Constrain	Description
Plan_id (primary key)	Integer	Min :1, Max:1	Unique number for plans.
Plan_name	Varchar		Lists of available plans.
Duration	Varchar		How long the subscription can last.
Price	Decimal		The amount for each of the plans.

2.1.1.5 Data dictionary for Element: usersubs

Name	Data Type	Constrain	Description
Transaction_id (primary key)	Integer	Min :1, Max:1	Given unique Identification of the transaction.
User_id (primary key)	Integer		Reference from the first table.
Plan_id(primary key)	Integer		Adopted from the subscription table.
Payment_method	Varchar		The chosen payment option from all the available lists in the comboBox.
Payment_info	Varchar		The details of the payment method.
Transaction_date	Timestamp		Date the purchase is made.

2.2 MySQL database design (Relational database)

2.2.1 Conceptual diagram



USER PROFILE

PK	User ID	Integer
Key	Nickname	Varchar (20)
Key	Birthday	Date (YYYY-MM-DD)
Key	Age	Integer
Key	Password	Varchar (255)
Key	Email Address	Varchar (100)
Key	Phone Number	Varchar (15)

TRANSACTIONS

PK	Transaction ID	Integer
FK	User ID	Integer
FK	Plan ID	Integer
Key	Payment Methods	Varchar (50)
Key	Payment Information	Varchar(50)
Key	Transaction Date	Time Stamp

CONVERSATIONS

PK	Message ID	Integer
FK	Match ID	Integer
FK	Sender ID	Integer
Key	Text Messages	Text
Key	Message Date	Time Stamp

PLANS

PK	Plan ID	Integer
Key	Duration	Varchar (25)
Key	Plan Name	Varchar (25)
Key	Price	Decimal

MATCH MAKING

PK	Match ID	Integer
FK	User ID	Integer
FK	Matched User ID	Integer
Key	Match Date	Time Stamp

2.2.2 Description

The conceptual framework or model of the mySQL database handler is shown in this diagram. It will ensure that the name and data type are implemented correctly once the database has been built and the other component has been added and chosen on the workbench. After that, the user is linked to the main program, where their credentials are immediately saved in the appropriate table if they enter them. This means that it will only take data that has been altered, particularly by the administrator or data entered the simulation. Every table has a function on the application's main page. We will observe the relationship as we proceed, as well as how and why they are connected. Five (5) tables make up the complete database, called logindb: bumbleusers, subsfeat, usermessages, userpreference, and usersubs.

2.2.3 Purpose of Tables

2.2.3.1 Purpose of bumbleusers Table

The information about the user's logged-in details is stored in this six-column table. Personal information like age, birthday, password, email address, and phone number will be obtained. Additionally, it ensures that the password is automatically hashed before being saved to the database, making it extremely secure. Only the administrator has the ability to change or modify the rows in these columns. Before creating an account, users must also fulfill certain requirements, such as the length and format of their passwords.

2.2.3.2 Purpose of userpreference Table

All of the user's preference information is stored in this four-column table. It will save the matched user (matched user ID) along with the action's time stamp after the user (user ID) swipes right. In this table, the user ID of the person logging in is the foreign key, and the match ID is the primary key.

2.2.3.3 Purpose of usermessages Table

Details of communications sent to other users in the account are stored by the user with five columns. After that user's (sender ID) and another user's (matching ID) IDs are matched, they will give each other an opportunity to get to know one another. Each message sent will be saved and stored in a system known as a message test. The message ID serves as the row's primary key, while the sender ID and matches ID are two foreign keys. Additionally, it will record the date that messages were sent, just like the other table.

2.2.3.4 Purpose of subsfeat Table

The purpose of this four-column table is to simply store all the plans that are available to the user (user ID from the premium to boost). It has a primary key called plan ID that will be imported into our next table. It also creates the plan name, which is the name of the plan, followed by duration, which is the length of time the users can enjoy the subscription, and finally, the price of the plans.

2.2.3.5 Purpose of usersubsTable

This six-column table is in charge of ensuring that every transaction pertaining to a subscription is documented. Consequently, the transaction ID serves as this table's primary key. Additionally, it has two primary keys: the plan ID from the table subsfeat and the user ID from the bumbleusers table. In addition to those three, the user's payment details, and method will be stored in this table. There are currently just four payment methods accepted by the system. Finally, for security purposes, every transaction is logged with a date.

.

2.2.4 Relations

From Table	To Table	Relation
bumbleusers	userpreference	A user to get a match and make a match.
userpreference	usermessages	A user from matched to reply to a message.
bumbleusers	usermessages	A user to make the first move or send a message.
subsfeat	usersubs	A plan ID to know what the user purchase.
bumbleusers	usersubs	To know who made the purchase.