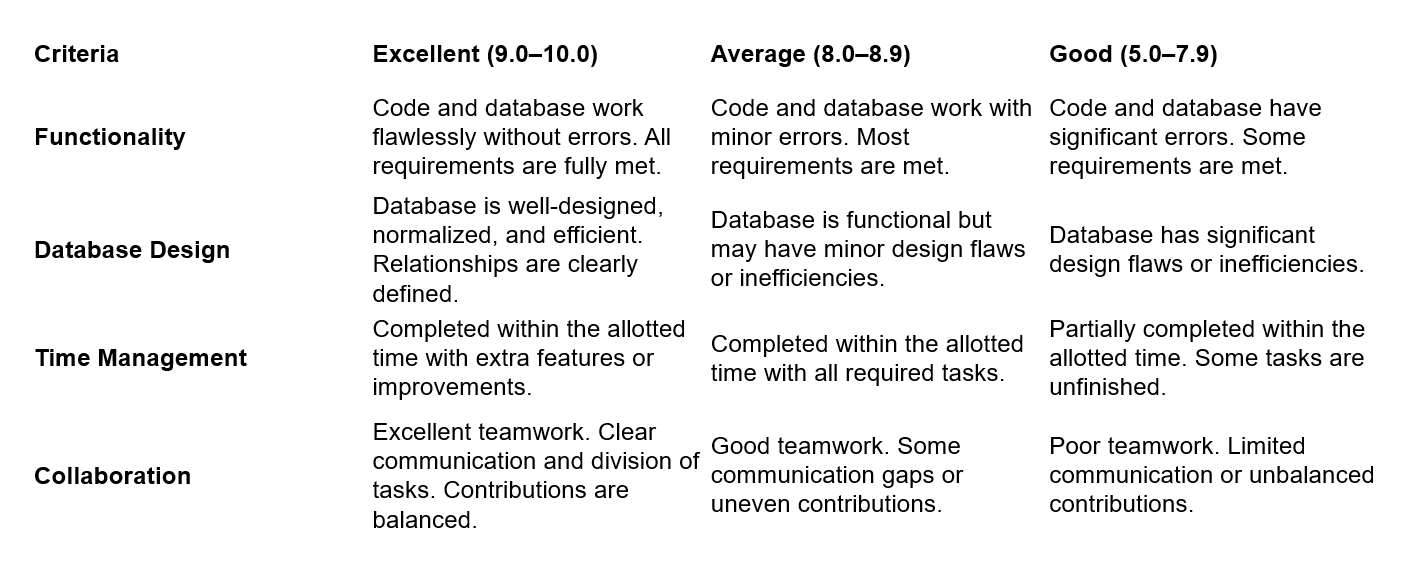
**Midterm Project in pairs:**

1. **Design a database based on your preferences.** Create a database system that reflects your interests or needs. Include all necessary tables, relationships, and constraints.
2. **Write an introduction for your system.** Provide a brief overview of the purpose and functionality of your database. Explain why you chose this particular design and how it aligns with your preferences.
3. **Design an Entity-Relationship Diagram (ERD).** Create a visual representation of your database structure, including entities, attributes, and relationships. Ensure the ERD is clear and follows standard conventions.
4. **Provide details of the database tables.** Describe the purpose of each table and its columns. Explain the relationships between tables and any constraints applied.
5. **Provide SQL statements.** Write CREATE TABLE statements to define the structure of your database. Include INSERT statements to populate your tables with sample data. Provide SELECT statements to demonstrate how data can be retrieved from your database.

**Deadline & Project Interview:** After Midterm Exam



Carmen Municipal College

A.Y. 2014-2025

**Title Here**

CC105 – Information Management

Midterm Project

LASTNAME, FIRSTNAME

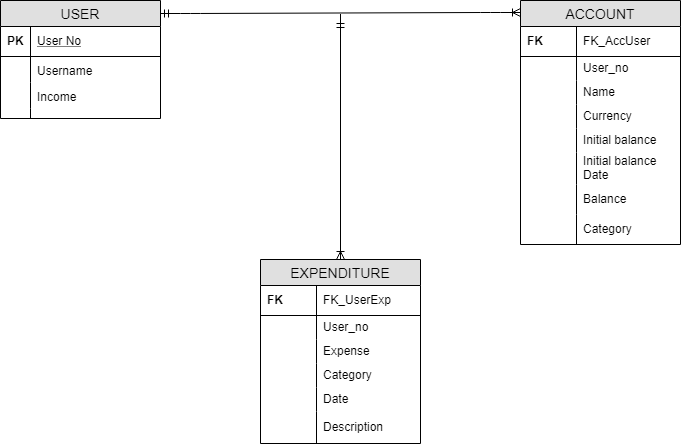
LASTNAME, FIRSTNAME

1. **Introduction**

Moneyger is a multi-platform mobile application that monitor the user’s daily expense. The designers target users are anyone who are conscious and would like to monitor and budget their daily expenses.

The users can *create* multiple account and can choose a category. Users can also filter their searches and the system *displays* the filtered information. The app is also capable of *updating* any information that the users would like to modify and can *delete* any data that are not useful to free some storage.

1. **Entity Relationship Diagram**



1. **Database**

Table Name: **USER**

This table contains the personal information of the user.

|  |  |  |
| --- | --- | --- |
| Fieldname | Type (size) | Description |
| user\_no | char(5) | Unique identifier of the user. |
| username | varchar2(50) | Name of the user. |
| income | number(6,2) | Income of the user. |

Table Name: **ACCOUNT**

This table contains the accounts of the user.

|  |  |  |
| --- | --- | --- |
| Fieldname | Type (size) | Description |
| acc\_no | char(5) | Unique identifier for the accounts of the user. |
| acc\_name | varchar2(50) | Account name of the user. |
| currency | varchar2(10) | Contains the currency. |
| initial\_balance | number(6,2) | Initial balance of the user. |
| Initial\_balance\_date | date | Initial balance date of the user |
| balance | number(6,2) | Current balance of the user. |
| category | varchar2(10) | Category of the account, ex. Cash or credit. |

Table Name: **EXPENDITURE**

This table contains the daily expense of the user.

|  |  |  |
| --- | --- | --- |
| Fieldname | Type (size) | Description |
| expense | number(6,2) | Expense of the user. |
| category | varchar2(15) | Category of the expense, ex. Food, clothing, travel, etc. |
| exp\_date | date | Expense date of the user. |
| description | varchar2(100) | Description of the expense. |

1. **Functionalities**

**Creating table:** this code shows the creation of the tables.

CREATE TABLE user (

user\_no char(5) PRIMARY KEY,

username varchar2(50),

income number(6,2)) engine = innoDB;

CREATE TABLE expenditure (

user\_no char(5),

expense number(6,2),

category varchar2(15),

exp\_date date,

description varchar2(100)

CONSTRAINT FK\_UserExp FOREIGN KEY (user\_no) REFERENCES user (user\_no) ON DELETE SET NULL ) engine = innoDB;

CREATE TABLE account (

user\_no char(5),

acc\_name varchar2(50),

currency varchar2(10),

initial\_balance number(6,2),

initial\_bal\_date date,

expense number(6,2),

balance number(6,2) AS (initial\_balance - expense),

category varchar2(10),

CONSTRAINT FK\_UserAcc FOREIGN KEY (user\_no) REFERENCES users (user\_no) ON DELETE SET NULL ) engine = innoDB;

**Inserting Users Information:** this code represents the user’s information input.

INSERT INTO users VALUES ( '00001', 'Tony Stark', 1005.00 );

**Inserting Account:** this code represents the creation of users account.

INSERT INTO account (user\_no, acc\_name, currency, initial\_balance, initial\_bal\_date, expense, category) VALUES ('00002', 'BDO Credit Card', 'USD', 9090.50, '01-25-2019', 0, 'Card');

INSERT INTO account (user\_no, acc\_name, currency, initial\_balance, initial\_bal\_date, expense, category) VALUES ('00002', 'Wallet Money', 'USD', 100.50, '02-25-2019', 1.5, 'Cash');

**Inserting Expense:** this code represents the users expense input.

INSERT INTO expenditure (expense, user\_no, category, exp\_date, description) VALUES ( 0.50, '00001', 'Food', '03-15-2019', 'Snacks');

INSERT INTO expenditure (expense, user\_no, category, exp\_date, description) VALUES ( 1.10, '00001', 'Food', '03-15-2019', 'Launch')

INSERT INTO expenditure (expense, user\_no, category, exp\_date, description) VALUES ( 2.00, '00001', 'Food', '03-15-2019', 'Dinner');

**Displaying:** this code represents the users filtered information.

SELECT \* FROM users;

SELECT \* FROM account WHERE category = 'Card';

SELECT category, COUNT (user\_no) FROM account GROUP BY category;

SELECT user\_no, acc\_name, category FROM account WHERE balance<=10000 ORDER BY user\_no;

SELECT \* FROM expenditure WHERE user\_no = '00001' ORDER BY expense;

SELECT \* FROM expenditure WHERE user\_no = '00002' AND category = 'Gadget' ORDER BY expense;