**Abstract:**

Detect Spam Base is a project aimed at developing a system that can detect and classify spam messages based on their content. The objective of this project is to create an efficient and accurate spam detection algorithm that can filter out unwanted messages and improve the overall user experience. The system will use machine learning techniques to analyse various features of messages such as keywords, frequency, and structure to determine their likelihood of being spam. This project is essential for organizations and individuals who receive a high volume of messages, as it can save time and reduce the risk of falling victim to malicious attacks.

**Tables:**

Based on the objective of the Detect Spam Base project, the following design requirements can be identified:

1. Message table: This table will contain all the messages that the system will analyse for spam detection. The attributes in this table could include message ID, sender, recipient, message content, date and time, and status (i.e., whether the message is spam or not).

2. Keyword table: This table will contain a list of keywords that the system will use to detect spam messages. The attributes in this table could include keyword ID, keyword name, and a flag indicating whether the keyword is commonly used in spam messages.

3. User table: This table will contain information about the users who will receive the messages. The attributes in this table could include user ID, name, email address, phone number, and any other relevant information.

4. Spam score table: This table will contain a score for each message, indicating the likelihood of the message being spam. The attributes in this table could include message ID, spam score, and any other relevant information.

5. Report table: This table will contain a record of all messages that were detected as spam by the system. The attributes in this table could include message ID, sender, recipient, reason for detection, and any other relevant information.

**CARDINALITIES AND CONSTRAINTS:**

1. Message table:

- Mapping cardinality: Many-to-One with User table (many messages can belong to one user)

- Constraints: None

- Key constraints: Primary key on message ID

2. Keyword table:

- Mapping cardinality: None

- Constraints: None

- Key constraints: Primary key on keyword ID, Unique constraint on keyword name

3. User table:

- Mapping cardinality: One-to-Many with Message table (one user can send or receive many messages)

- Constraints: None

- Key constraints: Primary key on user ID, Unique constraint on email address

4. Spam score table:

- Mapping cardinality: One-to-One with Message table (one message can have one spam score)

- Constraints: None

- Key constraints: Primary key on message ID, Unique constraint on message ID

5. Report table:

- Mapping cardinality: Many-to-One with Message table (many reports can be generated for one message)

- Constraints: None

- Key constraints: Primary key on report

Diagram

Description automatically generated

**ATTRIBUTES:**

1. Message table:

- Message ID (primary key)

- Sender

- Recipient

- Message content

- Date and time

- Status (e.g., spam or not spam)

2. Keyword table:

- Keyword ID (primary key)

- Keyword name

- Flag for commonly used in spam messages

3. User table:

- User ID (primary key)

- Name

- Email address (unique)

- Phone number

- Other relevant information (e.g., organization, department)

4. Spam score table:

- Message ID (primary key)

- Spam score

- Other relevant information (e.g., date and time of spam score calculation)

5. Report table:

- Report ID (primary key)

- Message ID

- Sender

- Recipient

- Reason for detection

**DDL OPERATIONS:**

1.Message Table:

CREATE TABLE message (

message\_id INT PRIMARY KEY,

sender VARCHAR(255),

recipient VARCHAR(255),

message\_content TEXT,

date\_time DATETIME,

status VARCHAR(255)

)

2.Keyword Table:

CREATE TABLE keyword (

keyword\_id INT PRIMARY KEY,

keyword\_name VARCHAR(255),

commonly\_used\_in\_spam BOOLEAN

);

3.User Table:

CREATE TABLE user (

user\_id INT PRIMARY KEY,

name VARCHAR(255),

email\_address VARCHAR(255) UNIQUE,

phone\_number VARCHAR(255),

other\_info TEXT

);

4.Spam score Table:

CREATE TABLE spam\_score (

message\_id INT PRIMARY KEY,

score NUMERIC(4,2),

other\_info TEXT

);

5.Report Table:

CREATE TABLE report (

report\_id INT PRIMARY KEY,

message\_id INT,

sender VARCHAR(255),

recipient VARCHAR(255),

reason\_for\_detection TEXT,

other\_info TEXT,

FOREIGN KEY (message\_id) REFERENCES message(message\_id)

);

**DML OPERATIONS:**

1.Message Table:

INSERT INTO message (message\_id, sender, recipient, message\_content, date\_time, status)

VALUES (1, 'lia@example.com', 'jay@example.com', 'Congratulations, you have won a prize!', '2023-04-30 12:00:00', 'spam');

INSERT INTO message (message\_id, sender, recipient, message\_content, date\_time, status)

VALUES (2, 'jay@example.com', 'lia@example.com', 'Thank you for your email. I will get back to you as soon as possible.', '2023-04-30 12:30:00', 'not spam');

INSERT INTO message (message\_id, sender, recipient, message\_content, date\_time, status)

VALUES (3, 'set@example.com', 'jay@example.com', 'URGENT: You have been selected for an exclusive offer!', '2023-04-30 13:00:00', 'spam');

INSERT INTO message (message\_id, sender, recipient, message\_content, date\_time, status)

VALUES (4, 'jay@example.com', 'spam@example.com', 'Please do not send me any more spam messages.', '2023-04-30 13:30:00', 'not spam');

INSERT INTO message (message\_id, sender, recipient, message\_content, date\_time, status)

VALUES (5, 'lia@example.com', 'spam@example.com', 'Please remove me from your mailing list.', '2023-04-30 14:00:00', 'not spam');

OUPUT:

Query OK, 5 rows affected

2.Keyword Table:

INSERT INTO keyword (keyword\_id, keyword\_name, commonly\_used\_in\_spam)

VALUES (1, 'Free', true);

INSERT INTO keyword (keyword\_id, keyword\_name, commonly\_used\_in\_spam)

VALUES (2, 'Congratulations', true);

INSERT INTO keyword (keyword\_id, keyword\_name, commonly\_used\_in\_spam)

VALUES (3, 'Exclusive offer', true);

INSERT INTO keyword (keyword\_id, keyword\_name, commonly\_used\_in\_spam)

VALUES (4, 'Prize', true);

INSERT INTO keyword (keyword\_id, keyword\_name, commonly\_used\_in\_spam)

VALUES (5, 'Win', true);

OUTPUT:

Query OK, 5 rows affected

3.User Table:

INSERT INTO user (user\_id, name, email\_address, phone\_number, other\_info)

VALUES (1, 'lia Smith', 'lia@example.com', '123-456-7890', 'Marketing Manager');

INSERT INTO user (user\_id, name, email\_address, phone\_number, other\_info)

VALUES (2, 'Jay ', 'jay@example.com', '555-555-5555', 'Sales Representative');

INSERT INTO user (user\_id, name, email\_address, phone\_number, other\_info)

VALUES (3, 'Spam Sender', 'set@example.com', null, null);

OUTPUT:

Query OK, 3 rows affected

4.Spam score Table:

INSERT INTO spam\_score (message\_id, score, other\_info)

VALUES (1, 0.8, 'Calculated by spam detection algorithm');

INSERT INTO spam\_score (message\_id, score, other\_info)

VALUES (3, 0.95, 'Calculated by spam detection algorithm');

OUPTUT:

Query OK, 2 rows affected

5.Report Table:

INSERT INTO report (report\_id, message\_id, sender, recipient, reason\_for\_detection, other\_info)

OUTPUT:

Query OK, 1 row affected