1. The modified model.py.

# Software Architecture and Design Patterns -- Lab 2 starter code

# Copyright (C) 2021 Hui Lan

from dataclasses import dataclass

from app import get\_session

@dataclass

class Article:

article\_id: int

text: str

source: str

date: str

level: int

question: str

class NewWord:

def \_\_init\_\_(self, username, word='', date='yyyy-mm-dd'):

self.username = username

self.word = word

self.date = date

class Reading:

def \_\_init\_\_(self, username, article\_id):

self.username = username

self.article\_id = article\_id

class User:

def \_\_init\_\_(self, username, password='12345', start\_date='2021-05-19', expiry\_date='2031-05-19'):

self.username = username

self.password = password

self.start\_date = start\_date

self.expiry\_date = expiry\_date

self.\_read = []

def read\_article(self, article):

session = get\_session()

reading=Reading(self.username,article.article\_id)

session.add(reading)

session.commit()

session.close()

2. The modified orm.py

# Software Architecture and Design Patterns -- Lab 2 starter code

# Copyright (C) 2021 Hui Lan

from sqlalchemy import Table, MetaData, Column, Integer, String, Date, ForeignKey

from sqlalchemy.orm import mapper, relationship

import model

metadata = MetaData()

articles = Table(

'articles',

metadata,

Column('article\_id', Integer, primary\_key=True, autoincrement=True),

Column('text', String(10000)),

Column('source', String(100)),

Column('date', String(10)),

Column('level', Integer, nullable=False),

Column('question', String(1000)),

)

users = Table(

'users',

metadata,

Column('username', String(100), primary\_key=True),

Column('password', String(64)),

Column('start\_date', String(10), nullable=False),

Column('expiry\_date', String(10), nullable=False),

)

newwords = Table(

'newwords',

metadata,

Column('word\_id', Integer, primary\_key=True, autoincrement=True),

Column('username', String(100), ForeignKey('users.username')),

Column('word', String(20)),

Column('date', String(10)),

)

readings = Table(

'readings',

metadata,

Column('id', Integer, primary\_key=True, autoincrement=True),

Column('username', String(100), ForeignKey('users.username')),

Column('article\_id', Integer, ForeignKey('articles.article\_id')),

)

def start\_mappers():

mapper(model.Article, articles)

mapper(model.NewWord, newwords)

mapper(model.User, users)

mapper(model.Reading, readings)

3.A detailed explanation for why your modification works

model.py：

创建session对象:

session = get\_session()

# 创建新Reading对象,把用户的名字和读的文章id储存进新对象:

reading=Reading(self.username,article.article\_id)

# 添加到session:

session.add(reading)

# 提交即保存到数据库:

session.commit()

# 关闭session:

session.close()

orm.py.：

#把表映射到类

mapper(model.Article, articles)

mapper(model.NewWord, newwords)

mapper(model.User, users)

mapper(model.Reading, readings)