الجامعة الإسلامية العالمية ماليزيا
INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA
يونيبرسيتي إسلام انتارابغسا مليسيا

(Company No. 101067-P)

Garden of Knowledge and Virtue

PREMIER INTERNATIONAL ISLAMIC RESEARCH UNIVERSITY

KULLIYYAH OF INFORMATION & COMMUNICATION TECHNOLOGY

SEMESTER 2, 2018/2019

INFO 2103 Database Programming

SECTION 3

E-Book Library System (EXODUS)

PREPARED BY

NAME	MATRIC NO.
MUHAMAD ARIF LUTFI BIN AZIZ	1315791
MUHAMAD KHAIRUL AZMI BIN KHAIRUDIN	1716803
MUHAMMAD LUQMANULHAKIM BIN SA'ARI	1813225
AFIFI SYAHMI BIN KAMAL-LUDIN	1710129
ADAM IZZUDDIN BIN KHALID	1627111

LECTURER

DR. ZAINATUL SHIMA ABDULLAH

Table of Contents

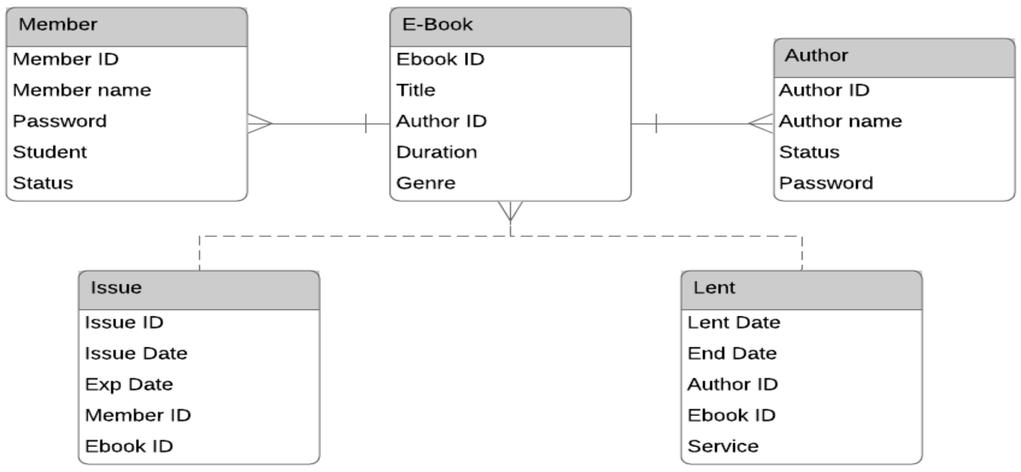
1.0 Introduction

In the eras of modernization, digitization of anything physical has been trending for the last 2 decades. Starting from a very simple digitization of letter into messaging apps to a point where physical books are turned into digital books, or e-books (or electronic books). Having the scent of paper on physical books may seduce book lovers, but the advantages of e-books lies in their mobility, accessibility and the most importantly, their availability to the public masses as a free material. However, availability of these e-books were often abused, by merging other material into one another and discarding the copyright of the e-books and stripping their authors' status of their work. Hence, our project, named as Exodus, focused on creating a database for managing all the e-books and exhibit all the advantages of e-books while still preserving its copyright and giving appreciation to authors for their work.

2.0 Objective

- To provide the simplest platform for the user to borrow and share book.
- To have the community that love to read and learn stuff.

3.0 Conceptual Database Design



4.0 Physical Database Design

Code below shows necessary table for the system which specifies all required attributes and constraints: -

```
-- CREATE TABLE AUTHOR ---------------------------------

CREATE TABLE member (
   memberid           VARCHAR2(10)          PRIMARY KEY,
   membername         VARCHAR2(30),
   pswd               VARCHAR2(8),
   stud               VARCHAR2(3),
   status             VARCHAR2(8));

-- CREATE TABLE AUTHOR ---------------------------------

CREATE TABLE author (
   authorid           VARCHAR2(10)          PRIMARY KEY,
   authorname         VARCHAR2(30),
   pswd               VARCHAR2(8),
   status             VARCHAR2(8));

-- CREATE TABLE EBOOK ----------------------------------

CREATE TABLE ebook (
   ebookid            VARCHAR2(10)          PRIMARY KEY,
   title              VARCHAR2(50),
   duration           NUMBER(3),
   genre              VARCHAR2(30));

-- CREATETABLE LENT ------------------------------------------------------

CREATE TABLE lent (
   lentid             NUMBER,
   lentdate           DATE,
   enddate            DATE,
   ebookid            VARCHAR2(10),
   authorid           VARCHAR2(10),
   service            VARCHAR2(10),
   CONSTRAINT lent_lentid_PK PRIMARY KEY (lentid),
   CONSTRAINT lent_ebookid_FK FOREIGN KEY (ebookid) REFERENCES ebook,
   CONSTRAINT lent_authorid_FK FOREIGN KEY (authorid) REFERENCES author);

-- CRAETE TABLE ISSUE ----------------------------------------------------

CREATE TABLE issue (
   issueid            NUMBER,
   issueddate         DATE,
   expdate            DATE,
   ebookid            VARCHAR2(10),
   memberid           VARCHAR2(10),
   CONSTRAINT issue_issueid_PK PRIMARY KEY (issueid),
   CONSTRAINT issue_ebookid_FK FOREIGN KEY (ebookid) REFERENCES ebook,
   CONSTRAINT issue_memberid_FK FOREIGN KEY (memberid) REFERENCES member);
```

The primary key of each of our table use sequence and trigger to auto generate their ID to avoid redundancy and inconsistency format from user input which will cause many difficulties later on as code given below: -

```
-- AUTO GENERATE MEMBER ID ----------------

CREATE SEQUENCE member_seq
   START WITH 1
   INCREMENT BY 1
   MAXVALUE 999
   NOCYCLE
   CACHE 20;

CREATE OR REPLACE TRIGGER memberid_trig
   BEFORE INSERT ON member
   FOR EACH ROW

BEGIN
   SELECT LPAD(member_seq.NEXTVAL, 3, '0')
   INTO :NEW.memberid
   FROM dual;
END;
/

-- AUTO GENERATE AUTHOR ID ----------------

CREATE SEQUENCE author_seq
   START WITH 1
   INCREMENT BY 1
   MAXVALUE 999
   NOCYCLE
   CACHE 20;

CREATE OR REPLACE TRIGGER authorid_trig
   BEFORE INSERT ON author
   FOR EACH ROW

BEGIN
   SELECT LPAD(author_seq.NEXTVAL, 3, '0')
   INTO :NEW.authorid
   FROM dual;
END;
/

-- AUTO GENERATE E-BOOK ID ---------------

CREATE SEQUENCE ebook_seq
   START WITH 1
   INCREMENT BY 1
   MAXVALUE 999
   NOCYCLE
   CACHE 20;

CREATE OR REPLACE TRIGGER ebookid_trig
   BEFORE INSERT ON ebook
   FOR EACH ROW
```

```
BEGIN
   SELECT LPAD(ebook_seq.NEXTVAL, 3, '0')
   INTO :NEW.ebookid
   FROM dual;
END;
/

-- AUTO GENERATE LENT ID -------------

CREATE SEQUENCE lent_seq
   START WITH 1
   INCREMENT BY 1
   NOMAXVALUE
   NOCYCLE
   CACHE 20;

CREATE OR REPLACE TRIGGER lentid_trig
   BEFORE INSERT ON lent
   FOR EACH ROW

BEGIN
   SELECT lent_seq.NEXTVAL
   INTO :NEW.lentid
   FROM dual;
END;
/

-- AUTO GENERATE ISSUE ID -------------

CREATE SEQUENCE issue_seq
   START WITH 1
   INCREMENT BY 1
   NOMAXVALUE
   NOCYCLE
   CACHE 20;

CREATE OR REPLACE TRIGGER issueid_trig
   BEFORE INSERT ON issue
   FOR EACH ROW

BEGIN
   SELECT issue_seq.NEXTVAL
   INTO :NEW.issueid
   FROM dual;
END;
/
```

5.0 Data Manipulation Language (DML)

We inserted 10 rows of data for each table for initial run test of the system as show in codes below: -

```
-- DATA TABLE MEMBER -------------------------------

INSERT INTO member (membername, pswd, stud, status)
VALUES ('Luqman Saari', '12345', 'YES', 'ACTIVE');

INSERT INTO member (membername, pswd, stud, status)
VALUES ('Arif Aziz', 'ILoveYou', 'YES', 'ACTIVE');

INSERT INTO member (membername, pswd, stud, status)
VALUES ('Asmak Nordin', '12345', 'NO', 'ACTIVE');

INSERT INTO member (membername, pswd, stud, status)
VALUES ('Laila Farhan', '12345', 'NO', 'INACTIVE');

INSERT INTO member (membername, pswd, stud, status)
VALUES ('Ilwa Chang', '12345', 'NO', 'ACTIVE');

INSERT INTO member (membername, pswd, stud, status)
VALUES ('Kang Seuk', '12345', 'NO', 'INACTIVE');

INSERT INTO member (membername, pswd, stud, status)
VALUES ('Hamadi Suhaimi', 'aabbcc12', 'YES', 'ACTIVE');

INSERT INTO member (membername, pswd, stud, status)
VALUES ('Liyana Aziz', '12345', 'NO', 'ACTIVE');

INSERT INTO member (membername, pswd, stud, status)
VALUES ('Hidayah Mat', '12345', 'NO', 'INACTIVE');


INSERT INTO member (membername, pswd, stud, status)
VALUES ('Aizat Ghuffar', 'AiAmFar', 'NO', 'ACTIVE');

-- DATA TABLE AUTHOR ------------------------------------

INSERT INTO author (authorname, pswd, status)
VALUES ('Suzanne Collins', '54321', 'ACTIVE');

INSERT INTO author (authorname, pswd, status)
VALUES ('Ryo Shirakome', '54321', 'ACTIVE');

INSERT INTO author (authorname, pswd, status)
VALUES ('Katharine Brooks', '54321', 'ACTIVE');

INSERT INTO author (authorname, pswd, status)
VALUES ('Ilwa Chang', '54321', 'ACTIVE');

INSERT INTO author (authorname, pswd, status)
VALUES ('Liyana Aziz', '54321', 'ACTIVE');
```

```
INSERT INTO author (authorname, pswd, status)
VALUES ('Stephen Pople', '54321', 'INACTIVE');

INSERT INTO author (authorname, pswd, status)
VALUES ('John Rowling', '54321', 'ACTIVE');

INSERT INTO author (authorname, pswd, status)
VALUES ('Lisa Kleypas', '54321', 'ACTIVE');

INSERT INTO author (authorname, pswd, status)
VALUES ('Aslam Ahmad', '54321', 'INACTIVE');

INSERT INTO author (authorname, pswd, status)
VALUES ('Fatin Mohamad', '54321', 'ACTIVE');

-- DATA TABLE EBOOK -------------------------------------------------------

INSERT INTO ebook (title, duration, genre)
VALUES ('The Hunger Games', 21, 'NOVEL SURVIVAL');

INSERT INTO ebook (title, duration, genre)
VALUES ('Commonplace Job to World Strongest', 14,'LIGHT NOVEL FANTASY');

INSERT INTO ebook (title, duration, genre)
VALUES ('The Scream', 21, 'NOVEL HORROR');

INSERT INTO ebook (title, duration, genre)
VALUES ('You Majored in What?', 30, 'EDUCATION');

INSERT INTO ebook (title, duration, genre)
VALUES ('Complete Physics for Cambridge IGCSE', 30,'EDUCATION');

INSERT INTO ebook (title, duration, genre)
VALUES ('Gravitational Implication Research', 30, 'REPORT');

INSERT INTO ebook (title, duration, genre)
VALUES ('Harry Potter and the Sorcerers Stone', 21, 'NOVEL FANTASY');

INSERT INTO ebook (title, duration, genre)
VALUES ('Devils Daughter', 21, 'NOVEL ROMANCE');

INSERT INTO ebook (title, duration, genre)
VALUES ('Fairy Tales', 14, 'NOVEL ONESHOT');

INSERT INTO ebook (title, duration, genre)
VALUES ('Philosophy of Islamization', 21, 'EDUCATION');

-- DATA TABLE LENT --------------------------------------------------------

INSERT INTO lent (lentdate, enddate, ebookid, authorid, service)
VALUES  (TO_DATE('01-MAY-2010'),  TO_DATE('01-MAY-2015'),  '001',  '001',
'FREE');

INSERT INTO lent (lentdate, enddate, ebookid, authorid, service)
VALUES  (TO_DATE('10-JUL-2012'),  TO_DATE('10-JUL-2020'),  '002',  '002',
'FREE');
```

```
INSERT INTO lent (lentdate, enddate, ebookid, authorid, service)
VALUES  (TO_DATE('06-APR-2014'),  TO_DATE('06-APR-2019'),  '003',  '001',
'FREE');

INSERT INTO lent (lentdate, enddate, ebookid, authorid, service)
VALUES  (TO_DATE('01-JAN-2015'),  TO_DATE('01-JAN-2025'),  '004',  '003',
'FREE');

INSERT INTO lent (lentdate, enddate, ebookid, authorid, service)
VALUES  (TO_DATE('03-MAR-2015'),  TO_DATE('03-MAR-2020'),  '005',  '006',
'FREE');

INSERT INTO lent (lentdate, enddate, ebookid, authorid, service)
VALUES  (TO_DATE('01-JAN-2017'),  TO_DATE('01-JAN-2022'),  '006',  '006',
'PREMIUM');

INSERT INTO lent (lentdate, enddate, ebookid, authorid, service)
VALUES  (TO_DATE('06-OCT-2018'),  TO_DATE('06-OCT-2025'),  '007',  '007',
'FREE');

INSERT INTO lent (lentdate, enddate, ebookid, authorid, service)
VALUES  (TO_DATE('01-DEC-2018'),  TO_DATE('01-DEC-2020'),  '008',  '008',
'PREMIUM');

INSERT INTO lent (lentdate, enddate, ebookid, authorid, service)
VALUES  (TO_DATE('02-JAN-2019'),  TO_DATE('02-JAN-2030'),  '009',  '009',
'FREE');

INSERT INTO lent (lentdate, enddate, ebookid, authorid, service)
VALUES  (TO_DATE('23-MAR-2019'),  TO_DATE('23-MAR-2025'),  '010',  '010',
'FREE');

-- DATA TABLE ISSUE -------------------------------------------------------

INSERT INTO issue (issueddate, expdate, ebookid, memberid)
VALUES (TO_DATE('15-SEP-2013'), TO_DATE('29-SEP-2013'), '002', '001');

INSERT INTO issue (issueddate, expdate, ebookid, memberid)
VALUES (TO_DATE('22-DEC-2013'), TO_DATE('12-JAN-2014'), '001', '005');

INSERT INTO issue (issueddate, expdate, ebookid, memberid)
VALUES (TO_DATE('13-MAR-2015'), TO_DATE('13-APR-2015'), '004', '004');

INSERT INTO issue (issueddate, expdate, ebookid, memberid)
VALUES (TO_DATE('20-MAR-2015'), TO_DATE('20-APR-2015'), '005', '004');

INSERT INTO issue (issueddate, expdate, ebookid, memberid)
VALUES (TO_DATE('01-NOV-2018'), TO_DATE('22-NOV-2018'), '007', '002');

INSERT INTO issue (issueddate, expdate, ebookid, memberid)
VALUES (TO_DATE('01-NOV-2018'), TO_DATE('22-NOV-2018'), '003', '002');

INSERT INTO issue (issueddate, expdate, ebookid, memberid)
VALUES (TO_DATE('24-DEC-2018'), TO_DATE('24-JAN-2019'), '006', '006');

INSERT INTO issue (issueddate, expdate, ebookid, memberid)
VALUES (TO_DATE('24-DEC-2018'), TO_DATE('24-JAN-2019'), '006', '006');
```

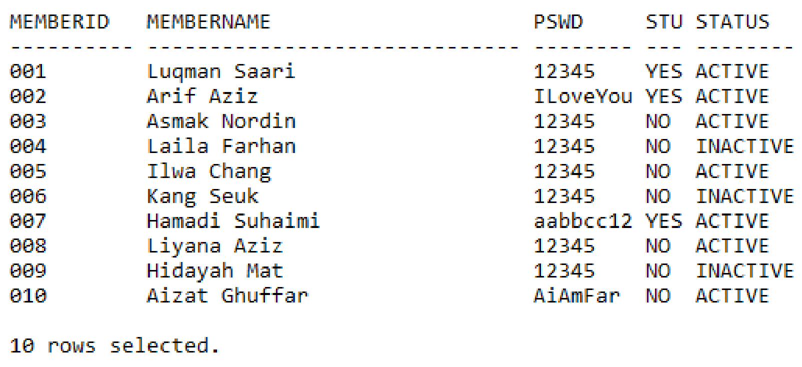
```
INSERT INTO issue (issueddate, expdate, ebookid, memberid)
VALUES (TO_DATE('14-JAN-2019'), TO_DATE('28-JAN-2019'), '002', '008');

INSERT INTO issue (issueddate, expdate, ebookid, memberid)
VALUES (TO_DATE('26-APR-2019'), TO_DATE('26-MAY-2019'), '010', '009');
```

After INSERT

a. Member table

Member table is the table that holds all the relevant information of the member of Exodus library. The primary key for this library is the member ID which is generated automatically using trigger and sequence in SQL.

```
SELECT * FROM member;
```



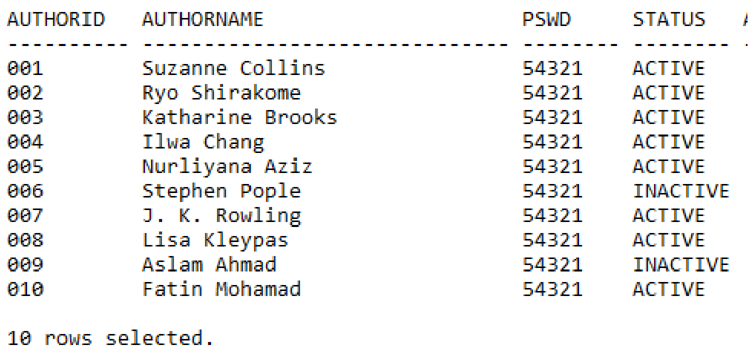
```
MEMBERID   MEMBERNAME                     PSWD     STU STATUS
---------- ------------------------------ -------- --- --------
001        Luqman Saari                   12345    YES ACTIVE
002        Arif Aziz                      ILoveYou YES ACTIVE
003        Asmak Nordin                   12345    NO  ACTIVE
004        Laila Farhan                   12345    NO  INACTIVE
005        Ilwa Chang                     12345    NO  ACTIVE
006        Kang Seuk                      12345    NO  INACTIVE
007        Hamadi Suhaimi                 aabbcc12 YES ACTIVE
008        Liyana Aziz                    12345    NO  ACTIVE
009        Hidayah Mat                    12345    NO  INACTIVE
010        Aizat Ghuffar                  AiAmFar  NO  ACTIVE

10 rows selected.
```

Screenshot 1: Display all data in table Member

b. Author table

Author table is the table that holds all the relevant information of authors that contributes their work to Exodus library. The primary key for this table was the author ID which also generated automatically using trigger and sequence in SQL.

```
SELECT * FROM author;
```



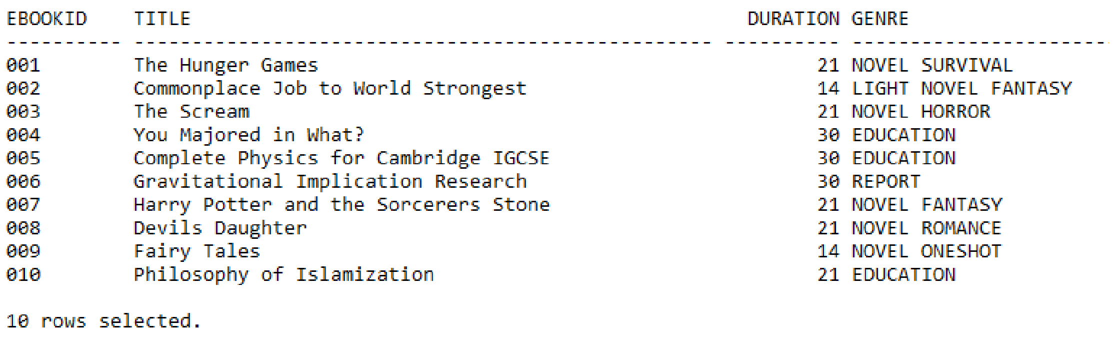
```
AUTHORID   AUTHORNAME                     PSWD     STATUS
---------- ------------------------------ -------- --------
001        Suzanne Collins                54321    ACTIVE
002        Ryo Shirakome                  54321    ACTIVE
003        Katharine Brooks               54321    ACTIVE
004        Ilwa Chang                     54321    ACTIVE
005        Nurliyana Aziz                 54321    ACTIVE
006        Stephen Pople                  54321    INACTIVE
007        J. K. Rowling                  54321    ACTIVE
008        Lisa Kleypas                   54321    ACTIVE
009        Aslam Ahmad                    54321    INACTIVE
010        Fatin Mohamad                  54321    ACTIVE

10 rows selected.
```

Screenshot 2: Display all data in table Author

c. eBook table

E-book table is the table that holds all the relevant information of the available books that are contributed by authors and how long they can be subscribed for free. It also has auto generated primary key, the ebook ID, using trigger and sequence in SQL.

```
SELECT * FROM ebook;
```



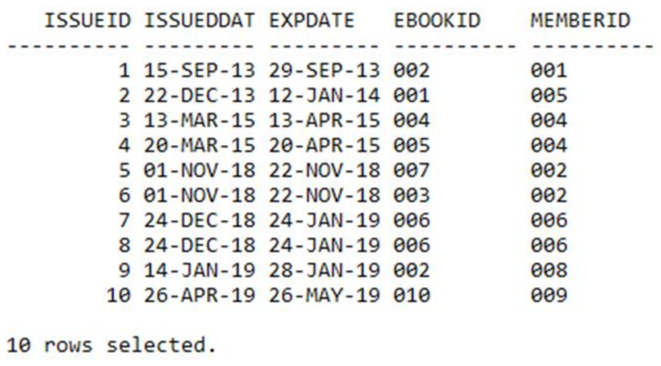
```
EBOOKID    TITLE                                              DURATION GENRE
---------- -------------------------------------------------- ---------- ----------------------
001        The Hunger Games                                          21 NOVEL SURVIVAL
002        Commonplace Job to World Strongest                        14 LIGHT NOVEL FANTASY
003        The Scream                                                21 NOVEL HORROR
004        You Majored in What?                                      30 EDUCATION
005        Complete Physics for Cambridge IGCSE                      30 EDUCATION
006        Gravitational Implication Research                        30 REPORT
007        Harry Potter and the Sorcerers Stone                      21 NOVEL FANTASY
008        Devils Daughter                                           21 NOVEL ROMANCE
009        Fairy Tales                                               14 NOVEL ONESHOT
010        Philosophy of Islamization                                21 EDUCATION

10 rows selected.
```

Screenshot 3: Display all data in table eBook

d. Issue table

Issue table is the table that connects the members and their subscription of a certain e-books. This table holds all the relevant information of when the e-book was issued to the member and their expiry date. It has its own primary key, which is Issue ID and holds 2 foreign key, member ID and ebook ID.

```
SELECT * FROM issue;
```



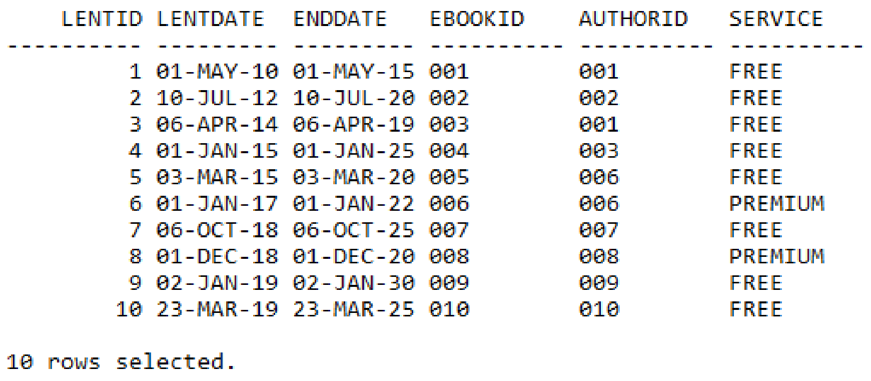
```
   ISSUEID ISSUEDDAT EXPDATE   EBOOKID    MEMBERID
---------- --------- --------- ---------- ----------
         1 15-SEP-13 29-SEP-13 002        001
         2 22-DEC-13 12-JAN-14 001        005
         3 13-MAR-15 13-APR-15 004        004
         4 20-MAR-15 20-APR-15 005        004
         5 01-NOV-18 22-NOV-18 007        002
         6 01-NOV-18 22-NOV-18 003        002
         7 24-DEC-18 24-JAN-19 006        006
         8 24-DEC-18 24-JAN-19 006        006
         9 14-JAN-19 28-JAN-19 002        008
        10 26-APR-19 26-MAY-19 010        009

10 rows selected.
```

Screenshot 4: Display all data in table Issue

e. Lent table

Lent table is the table that connect the authors table and their relevant works in e-book table. This table holds the information of when the author started contributing their works in Exodus library, and how long their work will be available in the library, and their current service type. All of the books will started out as free, and will only change to premium (paid subscription) if the author demands it. This table holds Lent ID as their primary key, and 2 foreign key, the author ID and ebook ID.

```
SELECT * FROM lent;
```

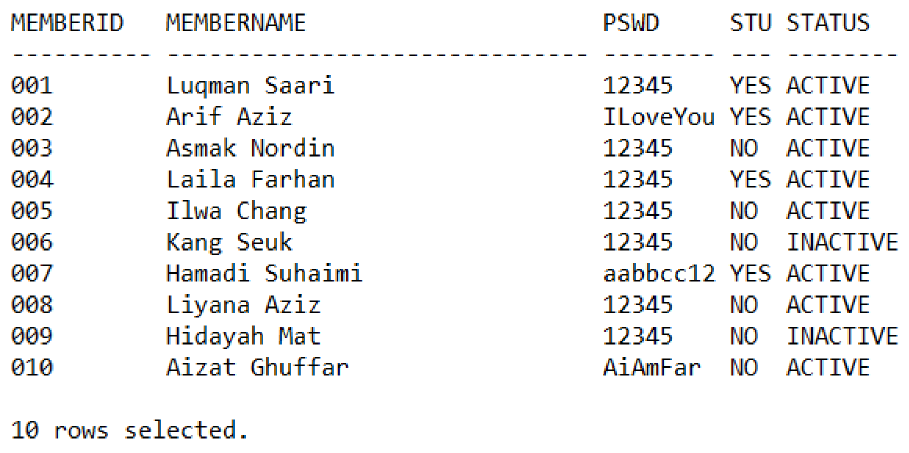


```
    LENTID LENTDATE  ENDDATE   EBOOKID    AUTHORID   SERVICE
---------- --------- --------- ---------- ---------- ----------
         1 01-MAY-10 01-MAY-15 001        001        FREE
         2 10-JUL-12 10-JUL-20 002        002        FREE
         3 06-APR-14 06-APR-19 003        001        FREE
         4 01-JAN-15 01-JAN-25 004        003        FREE
         5 03-MAR-15 03-MAR-20 005        006        FREE
         6 01-JAN-17 01-JAN-22 006        006        PREMIUM
         7 06-OCT-18 06-OCT-25 007        007        FREE
         8 01-DEC-18 01-DEC-20 008        008        PREMIUM
         9 02-JAN-19 02-JAN-30 009        009        FREE
        10 23-MAR-19 23-MAR-25 010        010        FREE

10 rows selected.
```

Screenshot 5: Display all data in table Lent

UPDATE

```
SQL> UPDATE member SET stud = 'YES', status = 'ACTIVE'
        WHERE membername = 'Laila Farhan';

1 row updated.
```

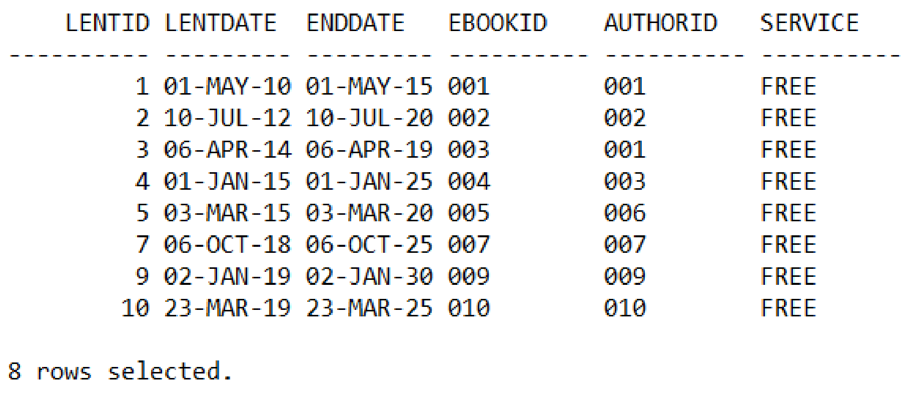


```
MEMBERID   MEMBERNAME                     PSWD     STU STATUS
---------- ------------------------------ -------- --- --------
001        Luqman Saari                   12345    YES ACTIVE
002        Arif Aziz                      ILoveYou YES ACTIVE
003        Asmak Nordin                   12345    NO  ACTIVE
004        Laila Farhan                   12345    YES ACTIVE
005        Ilwa Chang                     12345    NO  ACTIVE
006        Kang Seuk                      12345    NO  INACTIVE
007        Hamadi Suhaimi                 aabbcc12 YES ACTIVE
008        Liyana Aziz                    12345    NO  ACTIVE
009        Hidayah Mat                    12345    NO  INACTIVE
010        Aizat Ghuffar                  AiAmFar  NO  ACTIVE

10 rows selected.
```

Screenshot 6: Check table member for updated data

DELETE

```
SQL> DELETE FROM lent WHERE service = 'PREMIUM';

2 rows deleted.
```



```
    LENTID LENTDATE  ENDDATE   EBOOKID    AUTHORID   SERVICE
---------- --------- --------- ---------- ---------- ----------
         1 01-MAY-10 01-MAY-15 001        001        FREE
         2 10-JUL-12 10-JUL-20 002        002        FREE
         3 06-APR-14 06-APR-19 003        001        FREE
         4 01-JAN-15 01-JAN-25 004        003        FREE
         5 03-MAR-15 03-MAR-20 005        006        FREE
         7 06-OCT-18 06-OCT-25 007        007        FREE
         9 02-JAN-19 02-JAN-30 009        009        FREE
        10 23-MAR-19 23-MAR-25 010        010        FREE

8 rows selected.
```

Screenshot 7: Check table Lent for deleted data

6.0 Procedures

6.1 AddAuthor procedure

A procedure was created to add any new authors to the library. Since the author ID is automatically generated, we did not need to have any input for author ID. Unlike the other 2 procedures, this procedure is entirely optional since it is not needed if the author who wants to add more books for contribution already has his or her records available in the database.

Script

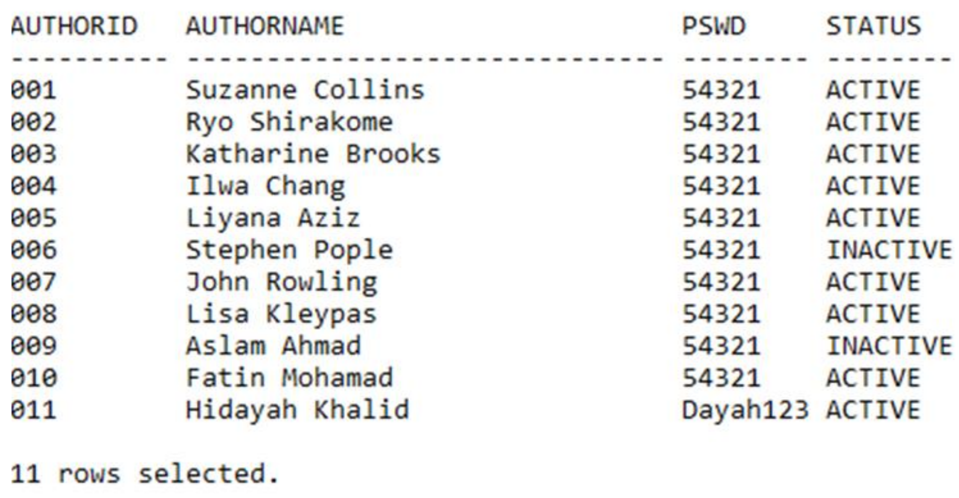
```
create or replace procedure AddAuthor(
   authorname IN author.authorname%TYPE,
   password IN author.pswd%TYPE,
   stat IN author.status%TYPE)
IS
BEGIN
   INSERT INTO AUTHOR ("AUTHORNAME","PSWD","STATUS")
   VALUES(authorname,password,UPPER(stat));
END;
/
```

Anonymous Block

```
ACCEPT name PROMPT 'Enter the Author name: ';
ACCEPT password PROMPT 'Create new password: ';
ACCEPT statuses PROMPT 'Status (ACTIVE/INACTIVE): ';

BEGIN
   AddAuthor('&name','&password','&statuses');
END;
/
```

After AddAuthor procedure

```
Enter the Author name: Hidayah Khalid
Create new password: Dayah123
Status (ACTIVE/INACTIVE): ACTIVE

PL/SQL procedure successfully completed.
```



```
AUTHORID   AUTHORNAME                     PSWD     STATUS
---------- ------------------------------ -------- --------
001        Suzanne Collins                54321    ACTIVE
002        Ryo Shirakome                  54321    ACTIVE
003        Katharine Brooks               54321    ACTIVE
004        Ilwa Chang                     54321    ACTIVE
005        Liyana Aziz                    54321    ACTIVE
006        Stephen Pople                  54321    INACTIVE
007        John Rowling                   54321    ACTIVE
008        Lisa Kleypas                   54321    ACTIVE
009        Aslam Ahmad                    54321    INACTIVE
010        Fatin Mohamad                  54321    ACTIVE
011        Hidayah Khalid                 Dayah123 ACTIVE

11 rows selected.
```

Screenshot 8: Check table Author for new row inserted using procedure

6.2 AddBook procedure

AddBook procedure is a procedure created if there is any new e-book available in the Exodus library that are available to the public. The e-book ID will be automatically generated. This procedure needed to be executed before the AddLent procedure.

Script

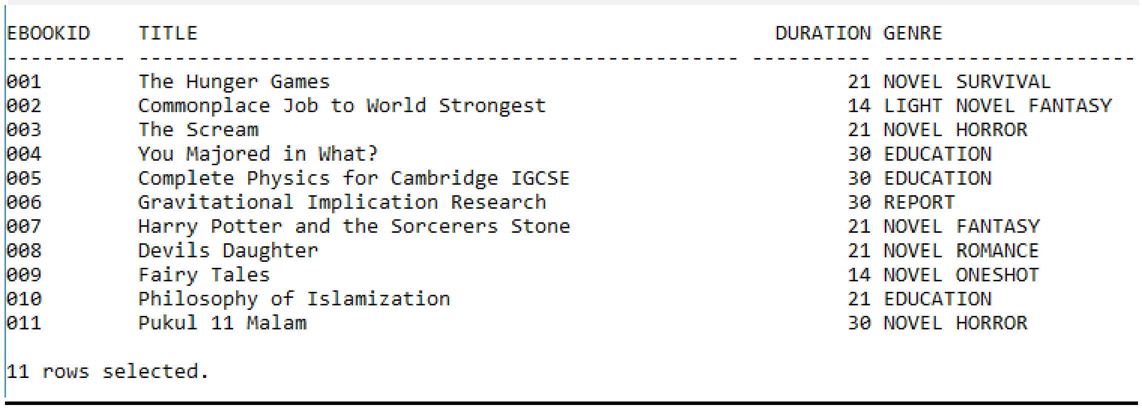
```
create or replace procedure AddBook(
   btitle IN ebook.title%TYPE,
   durations IN ebook.duration%TYPE,
   genres IN ebook.genre%TYPE)
IS
BEGIN
   INSERT INTO EBOOK ("TITLE","DURATION","GENRE")
   VALUES(btitle,durations,UPPER(genres));
END;
/
```

Anonymous Block

```
ACCEPT title PROMPT 'Enter Book title: ';
ACCEPT duration PROMPT 'Max borrow duration: ';
ACCEPT genre PROMPT 'Genre: ';

BEGIN
   AddBook('&title','&duration','&genre');
END;
/
```

After AddBook procedure

```
Enter the Book title: Pukul 11 Malam
Max borrow duration: 30
Genre: NOVEL HORROR

PL/SQL procedure successfully completed.
```



```
EBOOKID    TITLE                                                DURATION GENRE
---------- -------------------------------------------------- ---------- --------------------
001        The Hunger Games                                           21 NOVEL SURVIVAL
002        Commonplace Job to World Strongest                         14 LIGHT NOVEL FANTASY
003        The Scream                                                 21 NOVEL HORROR
004        You Majored in What?                                       30 EDUCATION
005        Complete Physics for Cambridge IGCSE                       30 EDUCATION
006        Gravitational Implication Research                         30 REPORT
007        Harry Potter and the Sorcerers Stone                       21 NOVEL FANTASY
008        Devils Daughter                                            21 NOVEL ROMANCE
009        Fairy Tales                                                14 NOVEL ONESHOT
010        Philosophy of Islamization                                 21 EDUCATION
011        Pukul 11 Malam                                             30 NOVEL HORROR

11 rows selected.
```

Screenshot 9: Check table eBook for new row inserted using procedure

6.3 AddLent procedure

AddLent procedure is the most important one since it connects the new books and their respective authors. The Lent ID are automatically generated, however the ebook ID and the author ID must be taken from existing sources in the database, which is why it is important to execute this procedure only after AddBook procedure was executed.

Script

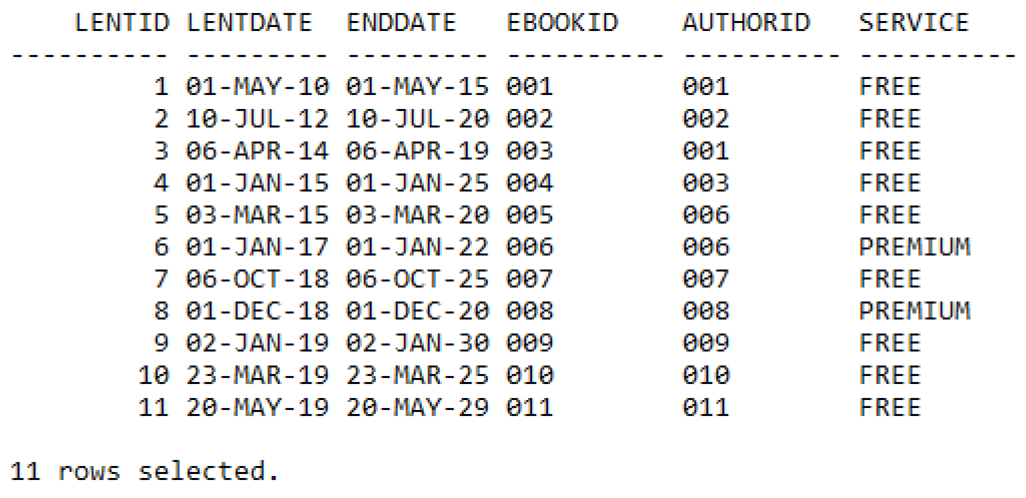
```
create or replace procedure AddLent(
   lent_date IN lent.lentdate%TYPE,
   end_date IN lent.enddate%TYPE,
   e_bookid IN lent.ebookid%TYPE,
   author_id IN lent.authorid%TYPE,
   serv IN lent.service%TYPE)
IS
BEGIN
   INSERT INTO LENT("LENTDATE", "ENDDATE", "EBOOKID", "AUTHORID",
      "SERVICE")
   VALUES (TO_DATE(lent_date), TO_DATE(end_date), e_bookid, author_id,
      serv);
END;
/
```

Anonymous Block

```
SET SERVEROUTPUT ON;
SET VERIFY OFF;
ACCEPT lentdate PROMPT 'Enter lend date: ';
ACCEPT enddate PROMPT 'Enter end lend date: ';
ACCEPT bookid PROMPT 'Enter Book ID: ';
ACCEPT authorid PROMPT 'Enter Author ID: ';
ACCEPT service PROMPT 'Enter service type (FREE/PREMIUM): ';

BEGIN
   AddLent('&lentdate','&enddate','&bookid', '&authorid', '&service');
END;
/
```

After AddLent procedure

```
Enter lent date: 20-May-29
Enter end lend date: 20-May-29
Enter Book ID: 011
Enter Author ID: 011
Enter service type (FREE/PREMIUM): FREE

PL/SQL procedure successfully completed.
```



```
    LENTID LENTDATE  ENDDATE   EBOOKID    AUTHORID   SERVICE
---------- --------- --------- ---------- ---------- ----------
         1 01-MAY-10 01-MAY-15 001        001        FREE
         2 10-JUL-12 10-JUL-20 002        002        FREE
         3 06-APR-14 06-APR-19 003        001        FREE
         4 01-JAN-15 01-JAN-25 004        003        FREE
         5 03-MAR-15 03-MAR-20 005        006        FREE
         6 01-JAN-17 01-JAN-22 006        006        PREMIUM
         7 06-OCT-18 06-OCT-25 007        007        FREE
         8 01-DEC-18 01-DEC-20 008        008        PREMIUM
         9 02-JAN-19 02-JAN-30 009        009        FREE
        10 23-MAR-19 23-MAR-25 010        010        FREE
        11 20-MAY-19 20-MAY-29 011        011        FREE

11 rows selected.
```

Screenshot 10: Check table Lent for new row inserted using procedure

7.0 Functions

7.1 chkAvailability function

Check whether the books are available or not. Since every books were contributed for a certain period of time, the member can check first their availability. It'll validate whether the issued date (current) is within the lending period from table Lent set by author.

Script

```
CREATE OR REPLACE FUNCTION chkavailable_sf(id IN VARCHAR2) RETURN
NUMBER IS
   validation       NUMBER;
   startdate        DATE;
   finishdate       DATE;

BEGIN
   BEGIN
      SELECT lentdate, enddate
      INTO startdate, finishdate
      FROM lent
      WHERE ebookid = id;

      IF SYSDATE > startdate AND SYSDATE < finishdate THEN
         validation := 1;
      ELSE
         validation := 0;
      END IF;

   EXCEPTION
      WHEN NO_DATA_FOUND THEN
         validation := 0;
   END;

   RETURN (validation);
END chkavailable_sf;
/
```

Anonymous Block

```
ACCEPT val PROMPT 'Enter the e-Book ID: ';

DECLARE
   bookid VARCHAR2(10) := '&val';
   validation NUMBER;
   vtitle ebook.title%TYPE;

BEGIN
   validation := chkavailable_sf(bookid);

   SELECT title INTO vtitleFROM ebook WHERE ebookid = bookid;
   IF validation = 1 THEN
      DBMS_OUTPUT.PUT_LINE('Requested book: ' || vtitle || ' (' ||
         bookid ||') | Status: AVAILABLE');
   ELSE
      DBMS_OUTPUT.PUT_LINE('Requested book: ' || vtitle || ' (' ||
         bookid ||') | Status: UNAVAILABLE');
   END IF;
END;
/
```

Example

```
Enter the e-Book ID: 009
Requested book: Fairy Tales (009) | Status: AVAILABLE

PL/SQL procedure successfully completed.
```

7.2 getExpDate function

This function is to retrieved the end date of issued e-book when member attempt to issue to reading. It'll will return the date value based on the duration set by the author lending their book. It uses formula of calculating current date plus by issue duration from table ebook.

Script

```
CREATE OR REPLACE FUNCTION getexpdate_sf (id in VARCHAR2) RETURN DATE
IS
   issueperiod ebook.duration%TYPE;
   expdate DATE;

BEGIN
   SELECT duration
   INTO issueperiod
   FROM ebook
   WHERE ebookid = id;

   expdate := (SYSDATE + issueperiod);

   RETURN (expdate);
END getexpdate_sf;
/
```

Anonymous Block

```
ACCEPT val PROMPT 'Enter the e-Book ID: ';
DECLARE
   ebookid VARCHAR2(10) := '&val';
   expireddate DATE;

BEGIN
   expireddate := getexpdate_sf(ebookid);
   DBMS_OUTPUT.PUT_LINE('Expired date for the issue is: ' ||
      expireddate);
END;
/
```

Example

```
Enter the e-Book ID: 011
Expired date for the issue is: 08-JUN-19

PL/SQL procedure successfully completed.
```

8.0 User Interfaces

In this topic, we discuss on a few sample of user interfaces for our e-Book Library System (Exodus). What we have manage to complete was only the interface design without any implementation as shown in the following: -

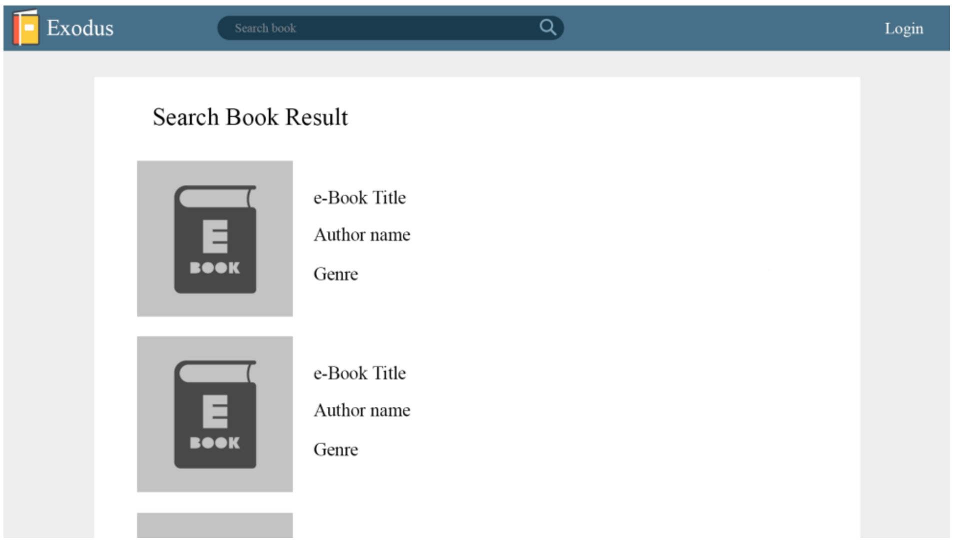


Image 1: Home page for list book that is currently available to be issued.

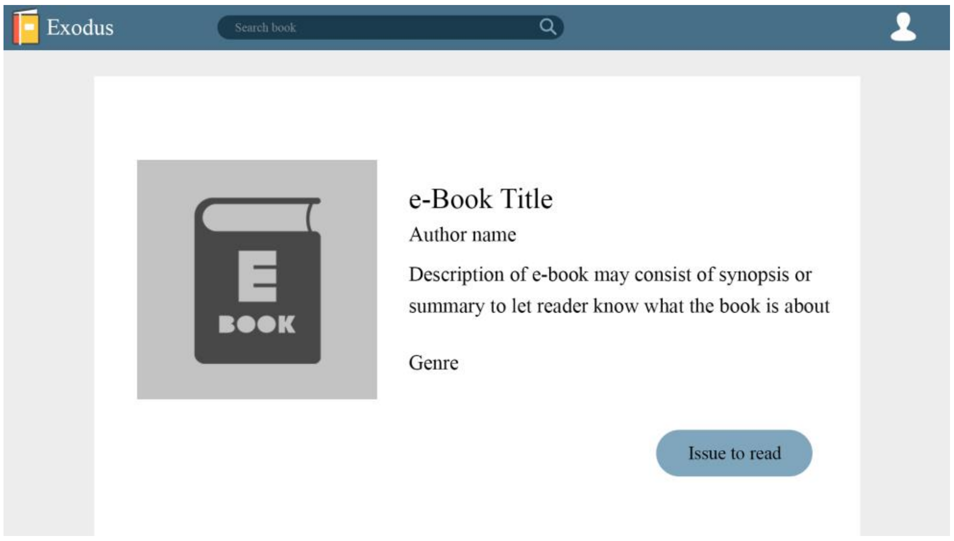

Exodus
Search book
e-Book Title
Author name
Description of e-book may consist of synopsis or summary to let reader know what the book is about
Genre
Issue to read

Image 2: e-Book Detail page where it shows all the detail regarding the book.

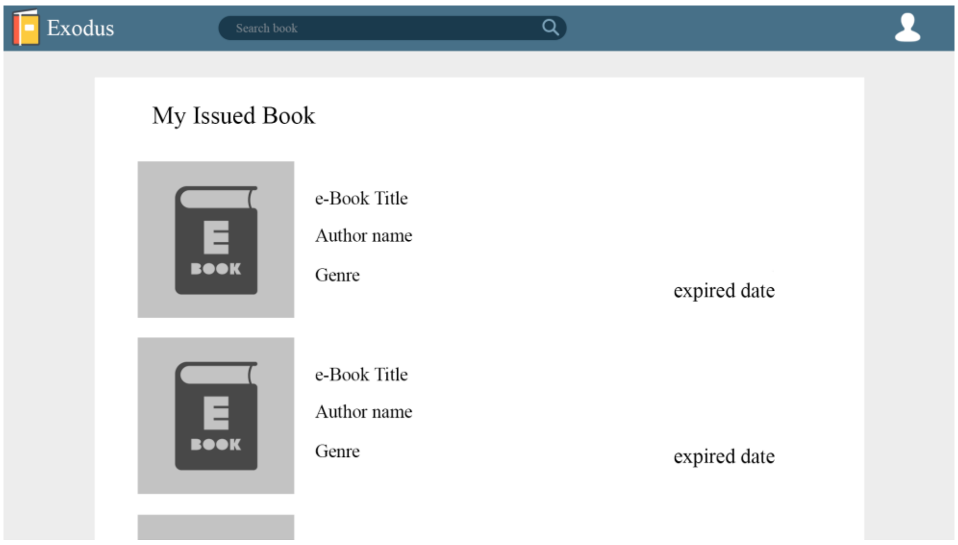


Image 3:Issued page where it shows the book that a member currently issueing.

9.0 Conclusion and Future Improvements

The world is changing towards digital and informational age. We are moving faster and faster towards virtual world while slowly eliminating the needs of physical items. Whenever there is change, the generation of people who lives in it, will often demand changes and simplicity to do things. This virtual and digital library is a perfect example of a platform that are in need in these current age. Applying simplicity along with ease of access, high mobility platform and wide availability will raise the demands of Exodus library. Be that as it may, there are still few improvements that we can instil for our Exodus.

9.1 Graphical User Interface (GUI)

An interface designed to be user-friendly and neat can bring out the best quality of our library. A GUI designed with simplicity yet highly functional features will be the bread and butter for Exodus. This will allow an execution of process to be done smoothly, thus attracting more members to try and explore Exodus as a new digital library platform.

9.2 Multiple platform accessibility

Exodus also needs to keep up with the rapid changes with the modernization of the current age. Connection Exodus database into multiple Internet of Things (IoT) platform will boost the accessibility exponentially. Device like tablets and smartphones are a must platform to have since it is a basic platform in the current age, however embedding Exodus into a platform like public transportation such as trains or flights, or personal transportation such as smart car or automobile will also maximize the mobility and accessibility of Exodus as a digital library. This will truly allow people to invest in reading books, almost anywhere and everywhere.

All in all, Exodus still have plenty of room to be improved and have the value to be commercialized as a product that can truly bring joy, ease and changes in the world of reading with the spirit of *Iqra'* as its core.