APPENDIX 1 CODE

VIEWS

```
/*/* ------------------------------------------------------
------------------------------------------------------ *

*              Script File Name : vw_TotalClassesPerMember

           *
*              Programmer Name : Tejas Dwarkaram

            *
*              Date : 2012.05.07

            *
*              Description : View total classes per member

            *
* ------------------------------------------------------
------------------------------------------------------ */
*/USE Suzi_Yoga_Studio
  GO
/*
CREATE VIEW vw_TotalClassesPerMember
AS
SELECT FoKOL
GO*/

/* ------------------------------------------------------
------------------------------------------------------ *
*              Script File Name : vw_ExerciseUsed

           *
*              Programmer Name : Tejas Dwarkaram

            *
*              Date : 2012.05.07

            *
*              Description : View the most commonly used

       *
* ------------------------------------------------------
------------------------------------------------------ */
CREATE VIEW vw_ExerciseUsed
AS
```

```
SELECT name, description, length_Of_Exercise,
number_Of_Times_To_Be_Done
FROM Exercises
GO

/* ------------------------------------------------------
------------------------------------------------- *
*              Script File Name : vw_Exercises

            *
*              Programmer Name : Tejas Dwarkaram

            *
*              Date : 2012.05.07

            *
*              Description : View title, name, description,
and time of exercises to be                              *
* ------------------------------------------------------
------------------------------------------------- */
CREATE VIEW vw_Exercises
AS
SELECT title, name, description,
number_Of_Times_To_Be_Done
FROM Exercise_Books JOIN Exercises
     ON Exercise_Books.book_ID = Exercises.book_ID
GO

/* ------------------------------------------------------
------------------------------------------------- *
*              Script File Name : vw_ClassAttendance

            *
*              Programmer Name : Tejas Dwarkaram

            *
*              Date : 2012.05.07

            *
*              Description : View the class attendance of
each
       *
* ------------------------------------------------------
------------------------------------------------- */
CREATE VIEW vw_ClassAttendance
AS
```

```
SELECT TOP 100 members.class_code,
class_sessions.day,class_sessions.time_Of_Session,
class_sessions.studio_Number
FROM members,class_sessions
WHERE members.class_code = class_sessions.class_code
ORDER BY members.class_code DESC

GO

SELECT *
FROM vw_ClassAttendance
```

TRIGGERS

```
/* ------------------------------------------------------
------------------------------------------------------ *
*              Script File Name : trg_underage

            *
*              Programmer Name : Tejas Dwarkaram

            *
*              Date : 2012.05.10

            *
*              Description : creating a trigger to check
the age of the new member
       *
* ------------------------------------------------------
------------------------------------------------------ */
USE Suzi_Yoga_Studio
GO

CREATE TRIGGER young_person
          ON Members AFTER INSERT
          AS IF UPDATE(date_of_birth)
          BEGIN
               DECLARE @age INT
               DECLARE @new_Birth DATE
               SET @new_Birth = (SELECT date_of_birth FROM
inserted)
               SET @age = (YEAR(GETDATE())-
(YEAR(@new_Birth)))
               IF(@age<16)
                    BEGIN
                         RAISERROR('You are not of the
appropriate age as yet.Only over 16 allowed.',16,10)
```

```
                        ROLLBACK TRANSACTION
                    END
            END
GO

/* ------------------------------------------------------
------------------------------------------------- *
*              Script File Name : trg_cancellation

            *
*              Programmer Name : Tejas Dwarkaram

            *
*              Date : 2012.05.10

            *
*              Description : creating a trigger to inform
suzi of the members that need to be informed if there  *
*                            is a cancellation

                *
* ------------------------------------------------------
------------------------------------------------- */
CREATE TRIGGER inform_cancel
     ON class_sessions AFTER UPDATE
     AS IF UPDATE(cancelled)
     DECLARE @cancelYes VARCHAR(1)
     DECLARE @classcode VARCHAR(6)
     SET @cancelYes = (SELECT cancelled FROM inserted)
     SET @classcode = (SELECT class_code FROM inserted)
     BEGIN
          IF(@cancelYes = 'Y')
               BEGIN
                    SELECT name, cell_Number, email_Address
                    INTO #tempTable
                    FROM members, class_sessions
                    WHERE members.class_code = @classcode
               END
     END
     PRINT 'The following members need to be informed of
the cancellation'
     SELECT *
     FROM #tempTable
GO
```

PROCEDURES

```
/* ------------------------------------------------------
------------------------------------------------------ *
*              Script File Name : sp_UpdateExerciseTimeUsed

       *
*              Programmer Name : Tejas Dwarkaram

            *
*              Date : 2012.05.03

            *
*              Description : Stored Procedure to updare the
number of times a certain exercise must be used      *
* ------------------------------------------------------
------------------------------------------------------ */

USE Suzi_Yoga_Studio
GO

CREATE PROCEDURE sp_UpdateExerciseTimeUsed(@book_ID
VARCHAR(30), @times INT)
     AS
     UPDATE Exercises
     SET number_Of_Times_To_Be_Done = @times
     WHERE book_ID = @book_ID
GO

/* ------------------------------------------------------
------------------------------------------------------ *
*              Script File Name : sp_Report

            *
*              Programmer Name : Tejas Dwarkaram

            *
*              Date : 2012.05.07

            *
*              Description : Get the report for a certin
class
            *
* ------------------------------------------------------
------------------------------------------------------ */
CREATE PROCEDURE sp_Report(@class_code VARCHAR(6))
AS
```

```
     DECLARE @day VARCHAR(10)
     DECLARE @studio_Number VARCHAR(10)
     DECLARE @name VARCHAR(45)
     DECLARE @cell VARCHAR(10)
     DECLARE @time VARCHAR(15)
     DECLARE @counter INT
     DECLARE @totalMembers INT
     DECLARE @auto VARCHAR (2)
     DECLARE @memNum INT
     DECLARE @mID INT

     SET @day = (SELECT day FROM class_sessions WHERE
@class_code = class_sessions.class_code )
     SET @time = (SELECT time_Of_Session FROM
class_sessions WHERE @class_code =
class_sessions.class_code )
     SET @studio_Number = (SELECT studio_Number FROM
class_sessions WHERE @class_code =
class_sessions.class_code )

          PRINT'YOGA CLASS REPORT:'
          PRINT'____________________'
          PRINT'Class Code:       ' + @class_code
          PRINT'Week Day:         ' + @day + '
     [Time:         ' + @time + ']'
          PRINT'Studio Number     ' + @studio_Number
          PRINT''
          PRINT'No. Member name            Contact number'
          PRINT'_____________________________________________'

          --SELECT name, cell_Number
          --FROM members
          --WHERE members.class_code = @class_code

     SELECT @totalMembers = COUNT(members.class_code)
     FROM members
     WHERE members.class_code = @class_code

     SET @auto = 1
     SET @counter = 0

     SELECT @mID = (members.member_Number)
     FROM members

     SELECT *
     FROM members
```

```
     WHERE member_Number = @mID

     WHILE @counter<@totalMembers
     BEGIN
          SELECT @name = members.name, @cell =
members.cell_Number
          FROM members
          WHERE members.class_code = @class_code
          Print @auto +'     ' + @name + '                '+
@cell
          SET @auto = @auto + 1
          SET @counter = @counter + 1
     END
GO

/* ------------------------------------------------------
------------------------------------------------- *
*              Script File Name : sp_DeleteBook

       *
*              Programmer Name : Tejas Dwarkaram

            *
*              Date : 2012.04.26

            *
*              Description : Stored Procedure to delete a
book record from the exercise books table            *
* ------------------------------------------------------
------------------------------------------------- */

CREATE PROCEDURE sp_DeleteBook(@book_ID VARCHAR(30))
     AS
     IF(@book_ID = (SELECT book_ID FROM vw_Exercises))
     BEGIN
          PRINT'The book that needs to be deleted cannot
be deleted at this time.'
     END
     ELSE
     BEGIN
          DELETE FROM Exercise_Books
          WHERE book_ID = @book_ID
     END
GO

/* ------------------------------------------------------
------------------------------------------------- *
```

```
*              Script File Name : sp_AddNewExercise

       *
*              Programmer Name : Tejas Dwarkaram

            *
*              Date : 2012.05.03

            *
*              Description : Stored Procedure to add a new
exercise record to the Exercises table
*
* ------------------------------------------------------
------------------------------------------------------ */
CREATE PROCEDURE sp_AddNewExercise(@bookID VARCHAR(30),
@name VARCHAR(100), @description VARCHAR(800), @length
TIME, @timesDone INT)
     AS
     INSERT INTO Exercises
     VALUES
(@bookID,@name,@description,@length,@timesDone)
GO
```

INDEXES

```
/* ------------------------------------------------------
------------------------------------------------------ *
*              Script File Name : ind_name

            *
*              Programmer Name : Tejas Dwarkaram

            *
*              Date : 2012.05.07

            *
*              Description : Index the title column

                *
* ------------------------------------------------------
------------------------------------------------------ */
USE Suzi_Yoga_Studio
GO

CREATE INDEX ind_name
ON exercises(name)
GO
```

DATABASE(inserting)

```
/* ------------------------------------------------------
------------------------------------------------------ *
*              Script File Name :
insert_all_of_the_sample_data
                                  *
*              Programmer Name : Tejas Dwarkaram

            *
*              Date : 2012.04.26

            *
*              Description : Inserting rows of data into
the tables
       *
* ------------------------------------------------------
------------------------------------------------------ */

USE Suzi_Yoga_Studio
GO

INSERT INTO class_sessions
VALUES('md02',1,'Monday',3,'03:30:00','N'),
     ('st40',3,'Wednesday',2,'16:00:00','N'),
     ('md01',2,'Tuesday',1,'09:30:00','N'),
     ('f32g',1,'Friday',3,'03:30:00','N')
GO

INSERT INTO
members(name,surname,date_of_birth,cell_Number,email_Addr
ess,class_code,studio_number,sessions_Per_Week)
VALUES('Tejas', 'Dwarkaram', '1993.11.22', '0829424982',
'tejas_dwarkaram@eml.cc','md02', 3, 3),
     ('Brandon', 'Rossouw', '1989.05.01', '0835734756',
'brandon.rossouw@gmail.com','f32g', 2, 5),
     ('Dimitri', 'Gonsalves', '1968.02.28', '0836785985',
'DimzG@mitrimail.com','md02', 3, 2),
     ('Devin', 'Botha', '1990.03.25', '0715454250',
'deviB@vinmail.com', 'st40', 1, 4),
     ('Albert', 'Rust', '1989.12.03', '0713254459',
'rust@telkomsa.com', 'st40', 4, 2),
     ('Nelio', 'Lucas', '1896.01.23', '0829654786',
'nelioL@rbgmail.com','f32g', 3, 6),
     ('Mamba', 'Samba', '1990.09.03', '0840021454',
'sambam@mamail.com','st40', 1, 3)
GO
```

```
INSERT INTO exercise_Books
VALUES('ks01','Yoga For Life','Bending'),
     ('sd01','Breathing New Light','Breathing
Techniques'),
     ('ks02','Mantra Healing','Self Realisation'),
     ('tj98','Meditation through the
Universe','Meditation')
GO

INSERT INTO exercises
VALUES('tj98','Rainbow Stretching','This entails the
usage of the lower back, in order to stretch the lower
      back area. This helps with improving posture, and
can help to relieve the syptoms of hunchback.'
      ,'01:00:00',25),
     ('ks02','Flamingo Meditation','This meditation
routine calms the mind, body and soul. It is done in a
     flamingo stance i.e. Standing on one foot at a time,
and raising ones hands above their head,
     in the form of praying.','00:03:00',4),
     ('ks02','Aathma Relief','Aathma Relief is done whilst
laying flat on ones body, joining ones feet together.
     By raising ur feet a little, without splitting them ,
you allow the flow of blood throughout the system.',
     '01:00:00',30),
     ('tj98','Splitting Bananas','Involving splits in yoga
practice allows for improved flexability of the body.
     By spontaneously doing splits in 4 directions, the
chances of arthirtis is lowered.',
     '02:00:00',6),
     ('sd01', 'Aum Relaxation','This is a meditation
technique that soothes the body and allows the body to
     communicate together as one.','00:30:00',5),
     ('ks01','Breath of life','This exercise, if done
correctly, opens the artiries and allows for intensiv
     deep breathing. This deep breathing helps with
circulation of air to the brain, as well increases the
flow
     of blood to the heart','01:00:00',15),
     ('sd01','Lotus Flower','Lotus is a sacred indian
flower. The purpose of this exercise is to loosen joints
     and enhance flexability which can strengthen the
muscles. Including biceps, triceps and calf muscles.',
     '00:45:00',10)
GO
```

DATABASe(create)

```
/* ------------------------------------------------------
------------------------------------------------------ *
*              Script File Name : Create_All_Base

            *
*              Programmer Name : Tejas Dwarkaram

            *
*              Date : 2012.04.26

            *
*              Description : Creating the database

            *
* ------------------------------------------------------
------------------------------------------------------ */

USE master
GO

IF EXISTS(SELECT name FROM master.dbo.sysdatabases
     WHERE name = 'Suzi_Yoga_Studio')
     DROP DATABASE Suzi_Yoga_Studio
     GO

CREATE DATABASE Suzi_Yoga_Studio
ON PRIMARY
(
     NAME = 'Suzi_Yoga_Studio_Data',
     FILENAME =
'C:\Users\Student\Downloads\Suzi_Yoga_Studio080512\Main_D
atabase\Suzi_Yoga_Studio.mdf',
     SIZE = 10,
     MAXSIZE = UNLIMITED,
     FILEGROWTH = 2
)
LOG ON
(

     NAME = 'Suzi_Yoga_Studio_Log',
     FILENAME =
'C:\Users\Student\Downloads\Suzi_Yoga_Studio080512\Main_D
atabase\Suzi_Yoga_Studio_Log.ldf',
     SIZE =10,
     MAXSIZE = UNLIMITED,
```

```
     FILEGROWTH = 2
)
GO

/* ------------------------------------------------------
------------------------------------------------------ *
*              Script File Name : inserting_tables

            *
*              Programmer Name : Tejas Dwarkaram

            *
*              Date : 2012.04.26

            *
*              Description : Inserting the tables

            *
* ------------------------------------------------------
------------------------------------------------------ */

USE Suzi_Yoga_Studio
GO

CREATE TABLE class_sessions
(
     class_code VARCHAR(6) NOT NULL,
     week INT NOT NULL,
     day VARCHAR(10) NOT NULL,
     studio_Number INT NOT NULL,
     time_Of_Session TIME NOT NULL,
     cancelled VARCHAR(1) DEFAULT 'N',
     CHECK(studio_Number IN(1,2,3,4)),
     CONSTRAINT prim_c_Code PRIMARY KEY(class_code)
)
GO

CREATE TABLE members
(
     member_Number INT NOT NULL IDENTITY(1, 1),
     name VARCHAR(45) NOT NULL,
     surname VARCHAR(45) NOT NULL,
     date_of_birth DATE NOT NULL,
     cell_Number VARCHAR(10) DEFAULT '0000000000',
     email_Address VARCHAR(90) NULL,
     class_code VARCHAR(6) NOT NULL,
     studio_number INT NOT NULL,
```

```
     sessions_Per_Week INT NULL,
     CONSTRAINT prim_memNum PRIMARY KEY(member_Number),
     CONSTRAINT uniq_Email UNIQUE(email_Address),
     CONSTRAINT for_Code FOREIGN KEY(class_code)
REFERENCES class_sessions(class_code)
)
GO

CREATE TABLE exercise_Books
(
     book_ID VARCHAR(30) NOT NULL,
     title VARCHAR(90) NOT NULL,
     exercise_Type VARCHAR(90) NOT NULL,
     CONSTRAINT prim_bookID PRIMARY KEY(book_ID)
)
GO


CREATE TABLE exercises
(
     book_ID VARCHAR(30) NOT NULL,
     name VARCHAR(100) NOT NULL,
     description VARCHAR(800) NOT NULL,
     length_Of_Exercise TIME NOT NULL,
     number_Of_Times_To_Be_Done INT NOT NULL,
     CONSTRAINT for_BookID FOREIGN KEY(book_ID) REFERENCES
Exercise_Books(book_ID)
)
GO
```

APPENDIX 2 USER_DOCUMENTATION

Table of Contents

Using SQL Management Studio

To begin SQL Management Studio, double click the desktop shortcut
(or if not found on desktop, locate the 'Microsoft SQL Server 2008' folder in the start menu)

Once opened connect to the '(local)' server, and then you should be presented with the following layout

To use any of the codes that are going to follow, click on "New Query"(usually found in the top left corner).
Please Note.
To execute a query press "F5"
And to end a query press "Alt" + "Break/Pause" together.

Using the created views

A view is used to view certain values of any table. What is represented is a kind of filter that contains data that is specified to be shown.

To use a view

A SELECT statement, is used on which we 'SELECT' everything from the 'view container'

The following needs to be typed in a 'New Query Editor Window'.

To view vw_Exercises

```
USE Suzi_Yoga_Studio
GO

SELECT *
FROM vw_Exercises
```

To view vw_ExerciseUsed

```
USE Suzi_Yoga_Studio
GO

SELECT *
FROM vw_ExerciseUsed
```

To view vw_ClassAttendance

```
USE Suzi_Yoga_Studio
GO

SELECT *
FROM vw_ClasAttendance
```

To view vw_TotalClassesPerMember

```
USE Suzi_Yoga_Studio
GO

SELECT *
FROM vw_TotalClassesPerMember
```

Using the created Stored Procedures

Stored procedures are also referred to as precompiled statements. This means that after one compilation, it is ready to be reused over and over again.

To use a stored procedure

We use the 'EXEC' or 'EXECUTE' statement to run a stored procedure. Most stored procedures have certain variables that must be provided for the procedure to function.

Eg.

A stored procedure for adding 2 numbers, will require 2 numbers. Variables are shown with "@" in the front of the variable name.

The format of a stored procedure is usually as follows;

<..procedure_name..>(@variable1, @variable2)

For example.

addNumbers(@num1, @num2)

So if we were to want to add 5 and 3, the code will look like this

```
USE Suzi_Yoga_Studio
GO

EXECUTE addNumbers @num1 = 3, @num2 = 5
GO
```

And upon execution(pressF5)

The answer '8' will be printed.

The following needs to be typed in a 'New Query Editor Window'.

To use sp_AddNewExercise

Variables required

@bookID which is the ID of the book
@name which is the name of the book
@description which is the description of the book
@length which is how long the exercise must be done for
@timesdone which is how many times the exercise must be done

What to type in order to execute the procedure

```
USE Suzi_Yoga_Studio
GO

EXEC sp_AddNewExercise @bookID=<..'bookID'..>, @name=<..'name'..>,
                       @description=<..'description'..>, @length=<..'length'..>,
                       @timesdone=<..'timestobedone'..>
GO
```

After execution a new record will be added to the Exercises table.

The following needs to be typed in a 'New Query Editor Window'.

To use sp_UpdateExerciseTimesUsed

Variables required

@bookID which is the ID of the book

@times which is how many times the exercise must be done

What to type in order to execute the procedure

```
USE Suzi_Yoga_Studio
GO

EXEC sp_UpdateExerciseTimeUsed  @bookID=<..'bookID'..>,
@timesdone=<..'timestobedone'..>
GO
```

After execution the number of times that the exercise will need to be done will be updated where the bookID is the value that you input.

The following needs to be typed in a 'New Query Editor Window'.

To use sp_DeleteBook

Variables required

@bookID which is the ID of the book

What to type in order to execute the procedure

```
USE Suzi_Yoga_Studio
GO

EXEC sp_DeleteBook @bookID=<..'bookID'..>
GO
```

After execution the procedure will first check whether or not the record exists as part of the Exercises view, if it does not, the record where the bookID is of the value provided, else the record will not be deleted.

The following needs to be typed in a 'New Query Editor Window'.

To use sp_Report

Variables required

@class_code which is the code of the class you want check

What to type in order to execute the procedure

```
USE Suzi_Yoga_Studio
GO

EXEC sp_Report @class_code = <..'code_of_the_class'..>
GO
```

After execution the procedure will print a report containing details about the class, including members and their cell numbers that are part of that class.

Author Notes

Name : Tejas

Surname : Dwarkaram

Date : 2012/05/10

Project Notes

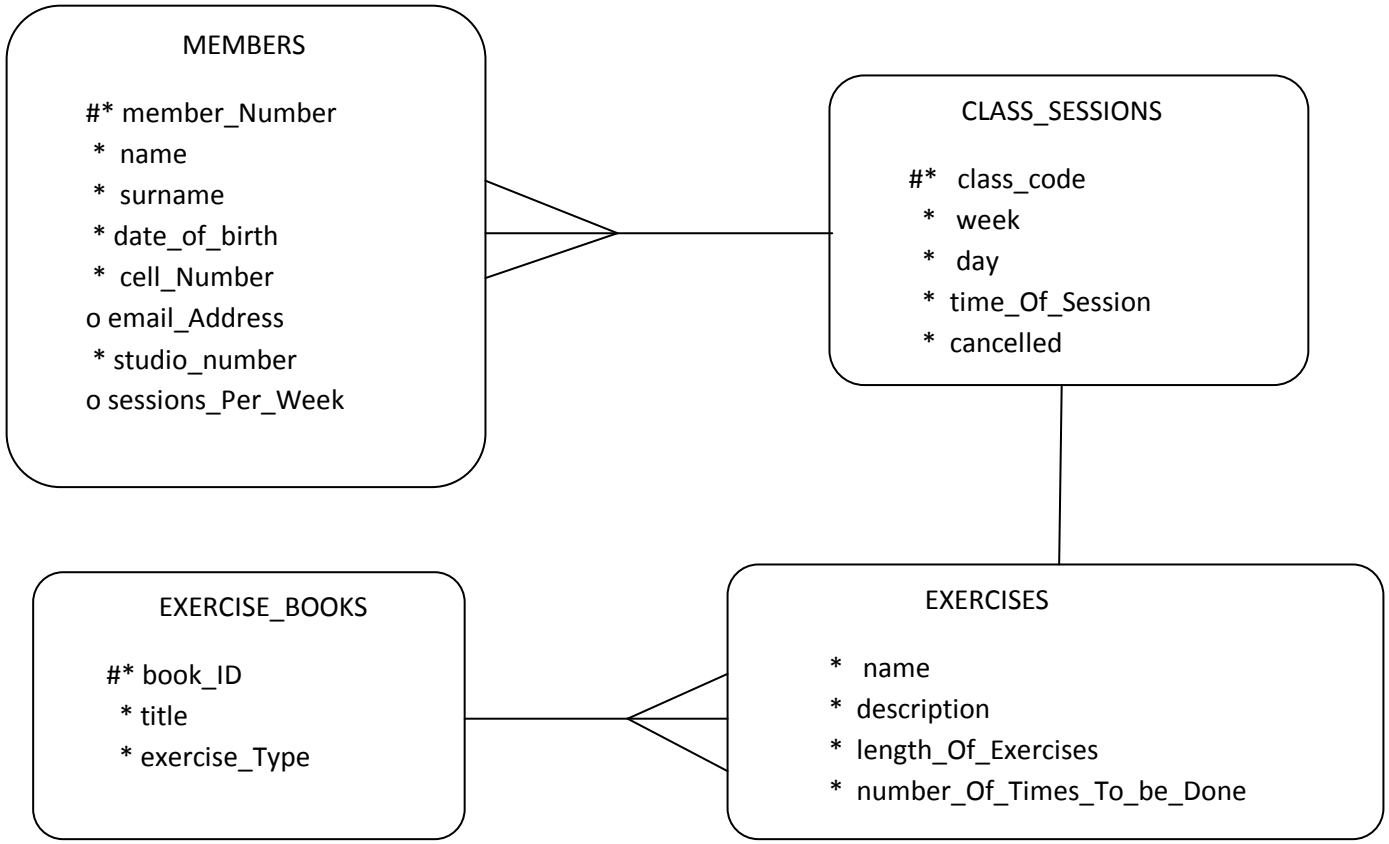
Purpose

The purpose of this project is to create a database as a student project. The database is created under scenario of a Yoga Studio.

Description

The database is for a Yoga Studio, named Suzi's Yoga Studio. This database will house information about the members, classes and exercises of Suzi's Yoga Studio. There are 4 tables that are utilized in this database. By making use of triggers, certain aspects of the tables are secured. Views have been created to cater for different criteria's. Stored procedures have been designed to carry out certain tasks easier, like adding records to the "Exercises" table.

ER Diagram


MEMBERS
#* member_Number
* name
* surname
* date_of_birth
* cell_Number
o email_Address
* studio_number
o sessions_Per_Week
CLASS_SESSIONS
#* class_code
* week
* day
* time_Of_Session
* cancelled
EXERCISE_BOOKS
#* book_ID
* title
* exercise_Type
EXERCISES
* name
* description
* length_Of_Exercises
* number_Of_Times_To_be_Done