**Test Driven Design for FLCKZ APP**

This document highlights the tests we did on the project(app) in order to improve the application.

The cases we testing in the test driven approach. Our testis are based on the following classes,in consideration to what this functions can achieve

We are testing the following Classes

1)Login

2)Register

3)Book

4)Order

**1) Login Testing**

a) The first tests we considered for login is if the application actually logs in.

**Test method**

Testing was done by accessing a user details from database to see if they match the data we have on our database

**Results of test**

A user was able to login,using our test method.,so the above test we used worked.

b) We check if a user that is not registered is not allowed to login.

**Test method**

The test method is similar to the one above.,we check for the user's entered details on our database.

**Results of test**

Our test tested was accepted cause if a user enters information which is not on our database(not registered),the user was not able to login..he had to register again.

c)We also try to logout of the application.

**Test method**

We tested this on the following scenarios if a user wants to logout he will exit from the application's interaction face to or back to our login page...Where he is required to login again.

**Results of Test**

This test failed cause when a user tries to logout basically nothing happens,they remained logged in...We looked at various solutions for this problem but failed to get a positive result on this test.

2) Register test

a )Run tests on whether uses with invalid emails are not allowed to register

**Test Method**

If a users email is off the wrong format or not an registered email the user must not be allowed to register.

**Results of Test**

This test was successful as a user was denied to freedom to register if their email was not of the correct format...this case was proven by multiple users.

b) If a person is registered they not allowed to register again.

**Test Method**

Basically this test method works by accessing registration database and checking whether a user is already registered..If they are we prevent them from registering again by printing a toast that tells them they already registered.

**Results of Test**

Our test was successful cause if a user was already registered,he was unable to register with the same details he had.

c) A person with a password shorter than 6 characters is not allowed to register.

**Test method**

The following test is done king the the amount of characters(String size) a user enters,if it less then six. We don't let the user register

**Results of Test**

Our test was accepted cause if a user entered a password which was less than six characters wrong he was not allowed to register into the application.

**3)Book testing**

a)If a booking is made it is well recorded,basically testing if a person is able to book.

**Test method**

A user books a table in the restaurant. He can either select a table of 2,family,friends or eat alone.

We test if a user can book a table by keeping record of user's choice,after a use books a certain table we store the user's table number(or table we reserved).

**Results of test**

After a lot of testing our test is accepted when a person enters or books a table the table they booked is stored in a database and recorded with the persons details.

b)We also run a test if the correct number of tables are booked(our app can trace if there is no more table to book).

**Test Method**

Basically we test this by looking into our database and accessing the number of tables we have in our restaurant,then if somebody books a table we keep decrementing the table size. The test here is that if database has no tables left..we Should alert the user.(

**Results of test**

This test was not yet passed by us,our table was not decrementing the number of tables available for booking so we are still doing further testing on it.

**4)Order Test**

a)We test whether a person is able to order a meal(clearly its whether a person can order food).

**Test Method**

If a user places an order we send them a toast message telling them what they ordered. If our application gives the correct ordered item,we consider it.

**Results of Test**

Our test was successful because everytime a user ordered items they received a correct message(alert) about the item they ordered.

b) If a user orders a meal.,can we store his order details so that we know what he ordered.(done to keep track of meals users buy).

**Test method**

We tested this on the fact that if a user orders a meal we store the type of order he makes.

And the test also checks if a user is also able to order multiple meals.

We test if a person orders the same thing twice it appears twice in the database as an increment.

**Results of test**

Our database is able to store a person's order ,like the person's order .So our first test case on ordering was accepted.

Our test works on multiple items if a person orders multiple meals our database shows the list of the items the person ordered

c Our third test however was not successful if a person orders something item twice it only appears once.,our class or function is not able to order the something twice