Software Testing CSE 225

Final Assessment

Restaurant Management system

Name: Ahmed Essam Mohamed Ramzy

ID: 17p8229

Department: CESS

Contents

1) Introduction:

What is software testing? It is the process of checking the functionality of a software application to discover whether the intended software have met the specified requirements or not and to see the defects of our system in order to publish our software in the market or for the buying facilities, as the process of testing can consume about 25-30 % from the time and cost of developing the software.

I thought of organizing our testing strategy by identifying the feature, its specification and then the test case used ; all after each other.

1) Logging in the system for the employee staff

1.1) The supportive models and assumptions for the extracted testable features:

Description of the feature: The first step in entering the system as every user using the system should login using the username and password given.

Assumption of the feature: I assume at this feature that all the users of the system already have a username and password.

Model: I will use the Decision table-testing model, its use to help test all combinations of conditions; it consists of two values conditions and actions.

Conditions	**Username**	F	T	F	T
	Password	F	F	T	T
Actions	**Excepted Result(E/S)**	E	E	E	S

Legends:

1) E: Error in both or either of the conditions.

2) S: Success in the login page and directed to the user page.

Scenarios for the decision table:

1) Success scenario: The user has entered correct username and password and the expected result will be the user will login to the user page.
2) Failed scenario: If the user enters either an incorrect username and password, the result will be an error message of either of them is incorrect.

1.2) Test case specifications:-

1.2.1) Preconditions:

1) The username characters are greater than 0.
2) The username can only contain letters(string).
3) The password length must be only integers which are greater than 8.

On failure of precondition: print " login failed ".

1.2.2) Post conditions:

As the general format is : if condition then effect 1 {else effect 2}

```
If( username>0 && password >8)
{
    If( Username& password matches in DB){
        Print (“login successfully”)
    }
}
Else
{
    Print (“login failed”)
}
```

1.2.3) Variables and operations

Variables/Inputs: Username of datatype string.

Password of datatype integer.

Operations: Login to the system.

1.3) Test case:

Login

Test Description(Scenario)	**Test data(Input)**	**Excepted results(output)**
It tests wrong username and password.	Username: Ahmeed Password: 129456789	The system prints ”login failed”
It tests empty username and password	User : Password:	The system prints ”login failed”
It tests that the username must be only letter & password must be only integers	User: Ahmed1 Password: 1234A56789	The system prints ”login failed”
It tests the correct username and password	User: Ahmed Password: 123456789	The system prints ”login successful”
It tests the correct username and password less than 8.	User: Ahmed Password: 123789	The system prints ”login failed”

2) Users staff different views to the system

2.1) The supportive models and assumptions for the extracted testable features:

Description of the feature: Every user in the system have a different view and access the system according to his job:-

1) Busboy and waiter and manager uses the small tabs.

2) Host uses the fixed terminals.

3) The kitchen staff uses the screens of orders.

Assumption of the feature: That there is a limited number of tabs/terminals/screens for the staff and according to the user job and the terminal he uses is free (not used by other employee) the system show him the home screen according to his role in the restaurant.

Model: I will use the cause effect graph for modeling this feature.

Causes:

1) User logged in as a waiter as c1.
2) User logged in as a host c2.
3) Terminal is free c3.
4) User logged in as a cook c4.
5) User logged in as a busboy c5.
6) User logged in as a manager as c6.

Effects:

1) Show system as waiter home screen as e1.
2) Print the terminal is used by another employee.

3) Show system as host home screen as e3.

4) Show system as kitchen staff home screen as e4.

5) Show system as busboy home screen as e5.

6) Show system as manager home screen as e6

#	1	2	3	4	5	6
C1	1	X	0	0	0	0
C2	0	X	1	0	0	0
C3	1	0	1	1	1	1
C4	0	X	0	1	0	0
C5	0	X	0	0	1	0
C6	0	X	0	0	0	1
E1	1	0	0	0	0	0
E2	0	1	0	0	0	0
E3	0	0	1	0	0	0
E4	0	0	0	1	0	0
E5	0	0	0	0	1	0
E6	0	0	0	0	0	1

2.2) Test case specifications:-

2.2.1) Preconditions:

1) The user has already logged in successfully.
2) The user already uses the terminal/ tab/ screen for showing his system view.

On failure of precondition: print “ Unauthorized page ”.

2.2.2) Post conditions:

As the general format is: if condition then effect 1 {else effect 2} so,

```
        If( login successfully)
    {
    If( User is waiter && uses tab)
    Print ("View waiter home screen")
    Else if ( User is manager && uses tab)
    Print ("View manager home screen")
    Else If( User is busboy&& uses tab)
    Print ("View busboy home screen")
    Else If( User is kitchen Staff&& uses screen)
    Print ("View kitchen Staff home screen")
    Else If( User is host && uses terminal)
    Print ("View host home screen")
    }
    Else
    {
    print " Unauthorized page ".
    }
```

2.2.3) Variables and operations

Variables/inputs: User/Account system**.**

Operations: Viewing different staff views for the system.

2.3) Test case:

User view to the system

Test Description(Scenario)	Test data(Input)	Excepted results(output)
Test the success of showing the waiter home screen.	1) User logged in as waiter successfully. 2) The tab is free to use	The system shows the home screen of the waiter.
Test the success of showing the host home screen.	1) User logged in as host successfully. 2) The Terminal is free to use.	The system shows the home screen of the host.
Test the success of showing the manager home screen.	1) User logged in as manager successfully. 2) The tab is free to use.	The system shows the home screen of the manager.
Test the success of showing the kitchen staff employee's home screen.	1) User logged in as any kitchen staff successfully. 2) The screen is free to use.	The system shows the home screen of the kitchen staff.
Test the success of showing the busboy home screen.	1) User logged in as any busboy successfully. 2) The tab is free to use.	The system shows the home screen of the busboy staff.
Test the abortion of viewing the home screen of any user wants to login.	1) User logged in as any staff successfully. 2) The Terminal/tab/screen is not free to use.	The system prints a message of "Unauthorized page".

3) Table status in the restaurant

3.1) The supportive models and assumptions for the extracted testable features:

Description of the features: This feature show the status of the table for the host, waiter staff in which the colors represent the tables status so that:-

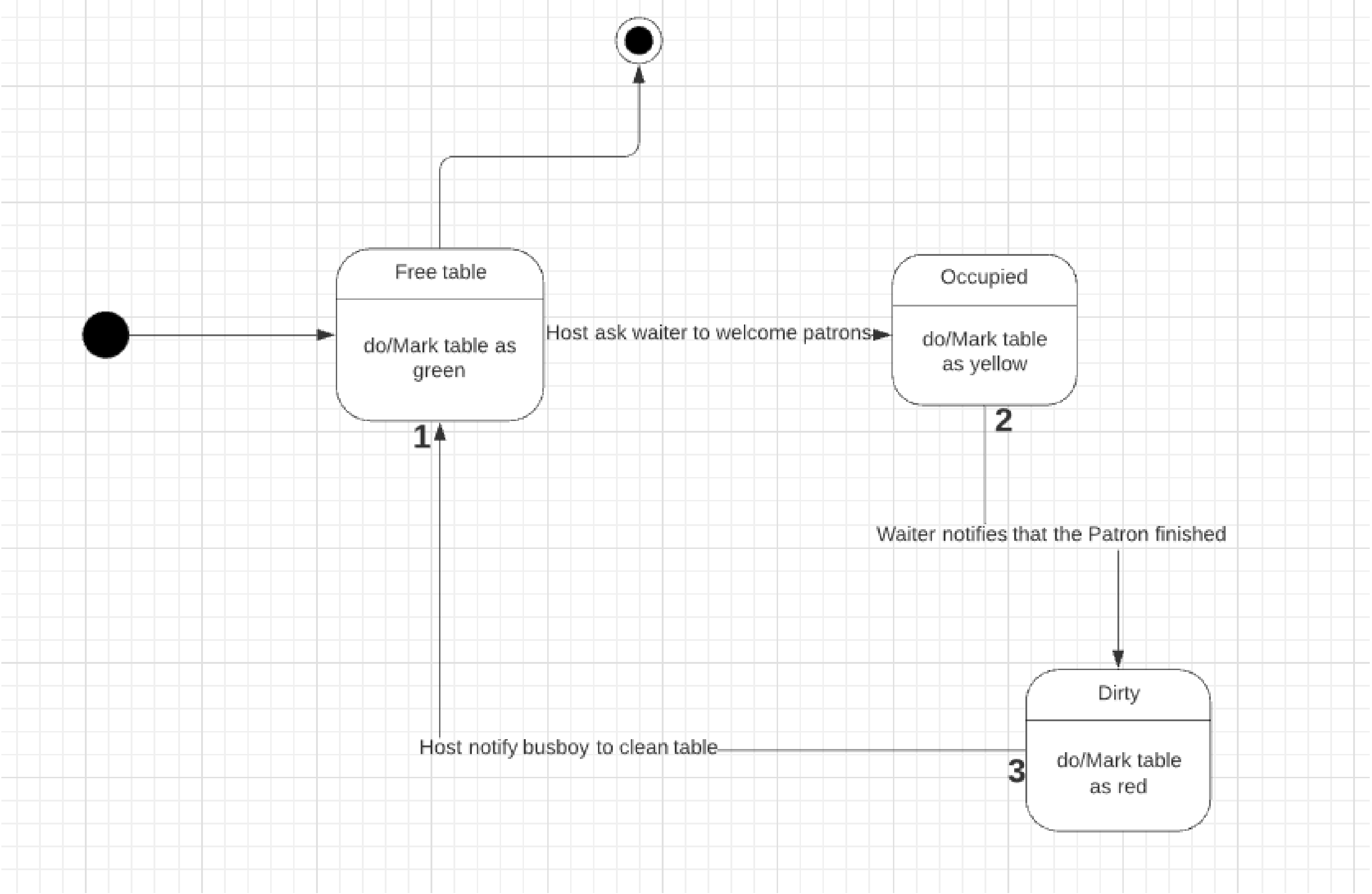
a. Host ask waiter to welcome the patrons(customers), if the tables colored green.

b. Waiter also notifies the host that the patron has finished.

c. Host let the busboy clean the tables if the tables colored red.

Assumption of the feature: I assumed at this feature that the system opens for a free table and then progress until the exit state.

Models: I will chose state transition testing to model this feature**.**


Free table
do/Mark table as green
Host ask waiter to welcome patrons
Occupied
do/Mark table as yellow
2
Waiter notifies that the Patron finished
Dirty
do/Mark table as red
3
Host notify busboy to clean table
1

3.2) Test case specifications:-

3.2.1) Preconditions:

1) Waiter/host logs in successfully.

1) Free table state stored in our system

On failure print “State cant be done”

Pre condition states:

1) State of table is free and show table color as greee.

3.3.2)Post conditions:

```
If ( free table ==true && patron.isWelcomed()&& table is green)

Then {

Mark table as yellow && free table==false

If (patron.isFinished()==true)

Then

{

Mark table as red && free table=false && busboy.Notified()

}

}

Else

{

Print “State cant be done”

}
```

Post condition states:

1) State of table is occupied and show table color as yellow.
2) State of table is Dirty and show table color as red.

3.3.3) Variables/Inputs:

1) Enum of states that will change according to the state of table of 3 states(ready,ocuupied,dirty).

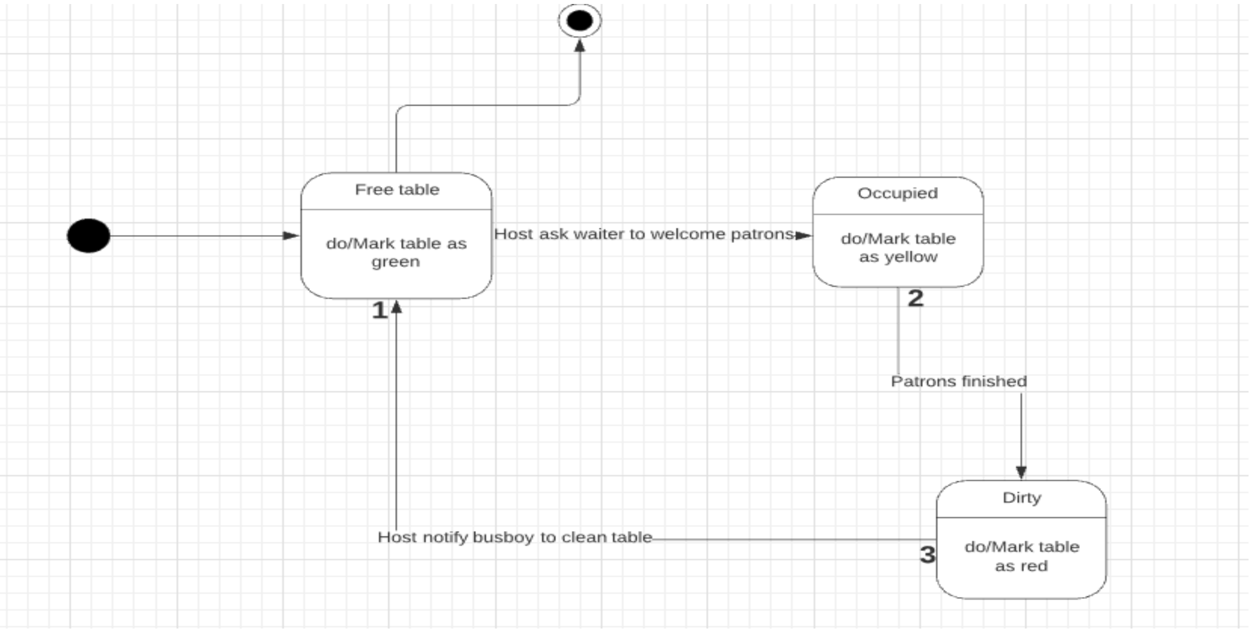
3.3.4) Operations:

1) Host ask waiter to welcome patrons.

2) Host notify busboy to clean the table.

3.3) Test case:

Table status(by path coverage)



Test Description(Scenario)	**Test data(Input)**	**Excepted results(output)**
Tests the system reaction on the tables, of the waiter and host home screens while changing the different status of tables.	➢ Waiter welcome the patron that enters the restaurant. ➢ Patron orders the food and eats. ➢ Host gets notified that the patrons finished. ➢ Host notify busboy to clean table and return to initial state. Test case: 1-> 2-> 3->1	1) Green. 2) Yellow. 3) Red. 4) Green.

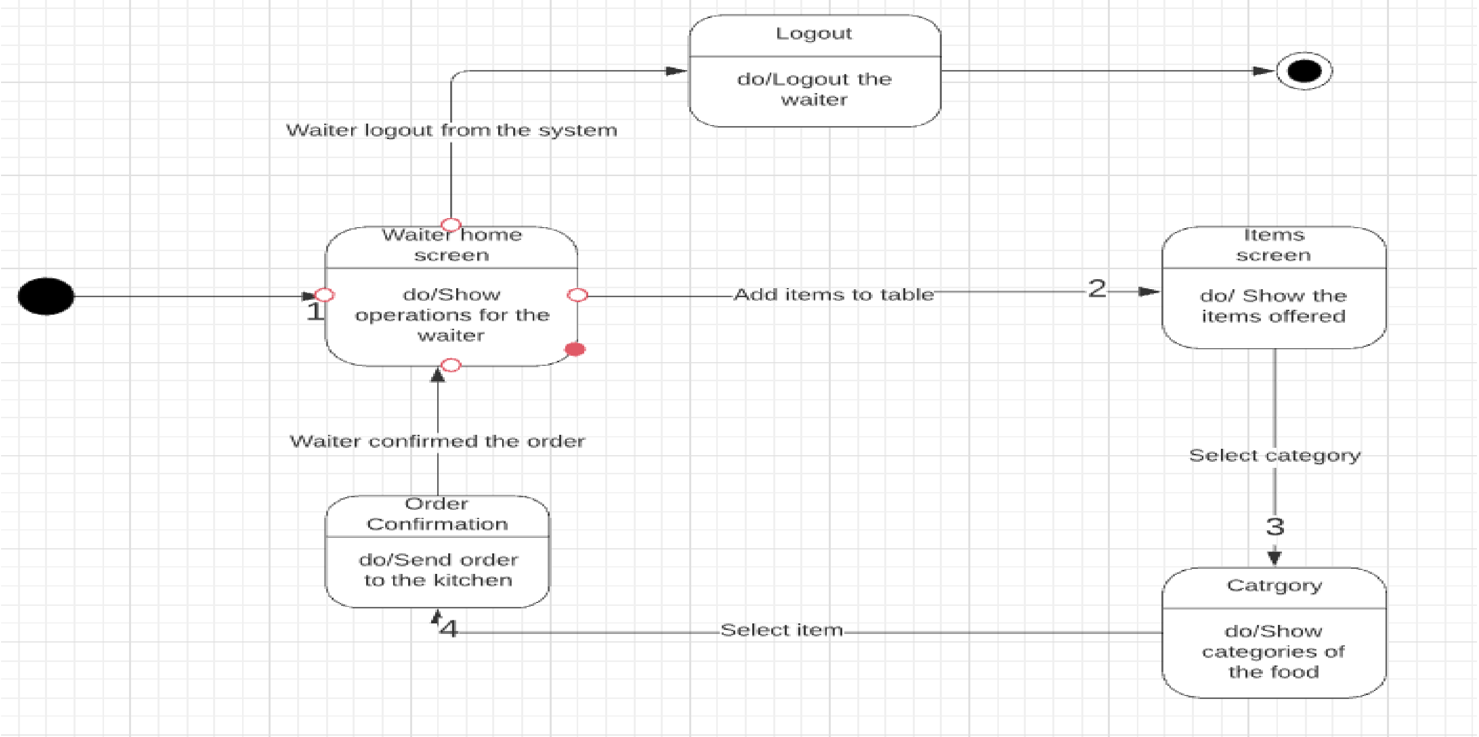
4) Waiter different operations states.

4.1) The supportive models and assumptions for the extracted testable features:

Description of the features: This feature show the states of the operations of the waiter home screen and the different operations he can do.

Assumption of the feature: I assumed at this feature that the waiter has only two operations in his screen, either to log out nor to proceed for confirming the order.

Models: I will chose state transition testing to model this feature**.**


Logout
do/Logout the waiter
Waiter logout from the system
Waiter home screen
do/Show operations for the waiter
1
Add items to table
2
Items screen
do/ Show the items offered
Select category
3
Waiter confirmed the order
Order Confirmation
do/Send order to the kitchen
4
Select item
Catrgory
do/Show categories of the food

4.2) Test case specifications:-

4.2.1) Preconditions:

1) Waiter is logged in successfully.

2) The tabs is free to use.

3) Waiter home screen is showed up to view the operations of the waiter.

On failure print “Operation cant be accessed”

4.4.2) Post conditions:

```
If ( getWaiterhomeScreen()==true && Waiter.additems())

Then {

Show items screen

If(getCategory()==true)

Then{

Show cateogries of food

If(getSelectItem()==true)

Then{

Send order to kitchen && return to home screen

}

If( Waiter.logOut())

Then{

Logout the waiter from the tab.

}

}

}
```

```
Else
{
    Print “Operation cant be acessed”
}
```

Post condition states:

1) Item screen state for waiter.

2) Category screen state for waiter.

3) Order confirmation screen state for waiter.

4) Logout the waiter screen.

4.4.3) Variables/inputs

1) Screen of waiter page that is changing across different operations.

4.4.4) Operations

1) Logout.

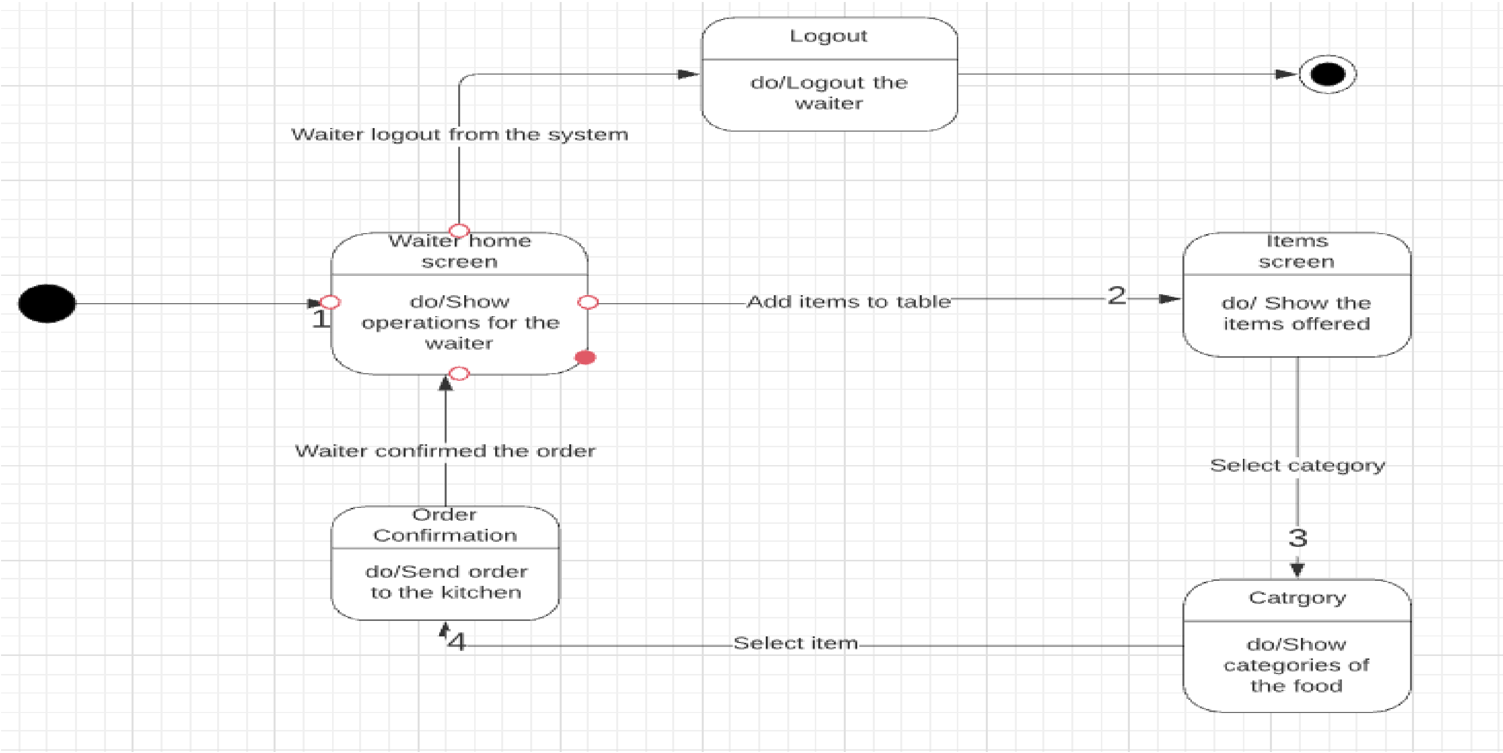
2) Show item offered for patron.

3) Show categories of food.

4) Send order to the kitchen.

4.3) Test case:

Waiter options(by path coverage)


Logout
do/Logout the waiter
Waiter logout from the system
Waiter home screen
do/Show operations for the waiter
1
Add items to table
2
Items screen
do/ Show the items offered
Select category
3
Catrgory
do/Show categories of the food
Select item
4
Order Confirmation
do/Send order to the kitchen
Waiter confirmed the order

Test Description(Scenario)	**Test data(Input)**	**Excepted results(output)**
It shows different states of the waiters screen	➢ Waiter chooses add items to table. ➢ Waiter selects category ➢ Waiter selects item ➢ Waiter confirmed item to the kitchen. ➢ Waiter chooses to logout. Test case: 1-> 2-> 3->4->1->5	1) Show items offered. 2) Show categories of the food. 3) Send order to the kitchen. 4) Waiter logs out from the system.

5) Updates and reorganization of data.

5.1) The supportive models and assumptions for the extracted testable features:

Description of the features: The updates and reorganization of data are triggered by logging in and out.

Assumption of the feature:

1) I assumed at this feature that the restaurant staff's data (waiter,busboy) would be updated if they both logged in and out ,but if the kitchen and host staff login once they will not require for frequent logouts.

2) I assumed also that the waiter data must be updated automatically after 10 seconds and don't need any both the log in and out for the update to occur.

Models: I will chose cause effect graph to model this feature**.**

Causes:

C1) Restaurant staff.

C2) Login.

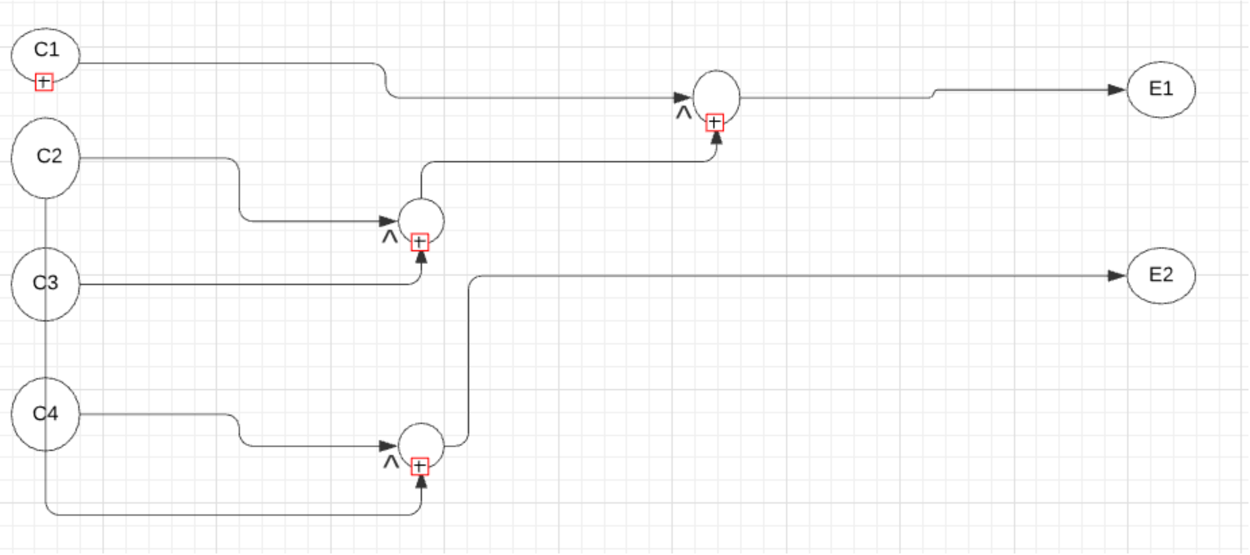
C3) logout.

C4) Kitchen and host staff.

Effects:

E1) Data is updated and reorganized in the system.

E2) The staff kept logged in and not require frequent logouts.



#	1	2
C1	1	0
C2	1	1
C3	1	0
C4	0	1
E1	**1**	**0**
E2	**0**	**1**

5.2) Test case specifications:-

5.2.1) Preconditions:

1) User is logged in to the system according to his profession.

1) There is available data for the employee’s staff.

2) Data in the database have been changed/ triggered according to the operations done.

On failure leave the data unchanged in the database

5.2.2) Post conditions:

```
If (User is logged in && Data.isavailable()==true && Data.changed())

Then

 {

Update data in the database

}

Else

{

Keep data same in the database

}
```

5.2.3) Variables/inputs:

1) Data of the employees.

5.2.4) Operations:

1) Writing and reading on the database.

2) Data is successfully updated in the database and in the employees screen.

5.3) Test case:

User data update

Test Description(Scenario)	**Test data(Input)**	**Excepted results(output)**
Test the updating of data for the normal restaurant staff	1) If the user logged in as a restaurant staff (manager and busboy). 2) The user is both logged in and logged out to the system successfully. 3) Data is available and triggered/changed.	The data will be updated in both the database and in the tabs/terminals of the Staff.
Test the updated data of both the kitchen and host staff	1) User logged in successfully. 2) The user staff logging in is either kitchen or host staff. 3) Data is updated/triggered.	The data will be updated in both the database and in the tabs/terminals of the Staff.

6) Manager operations for profiles:

6.1) The supportive models and assumptions for the extracted testable features:

Description of the features: The manager is able to create, modify and access profiles of the employee.

Assumption of the feature: I assumed at this feature that only the manager can manipulate with the profiles of the employees and not any staff other than him.

Models: I used cause effect graph for modeling this feature.

Causes:

C1) Employee is a manager

C2) Creation of the employee profiles page in the system

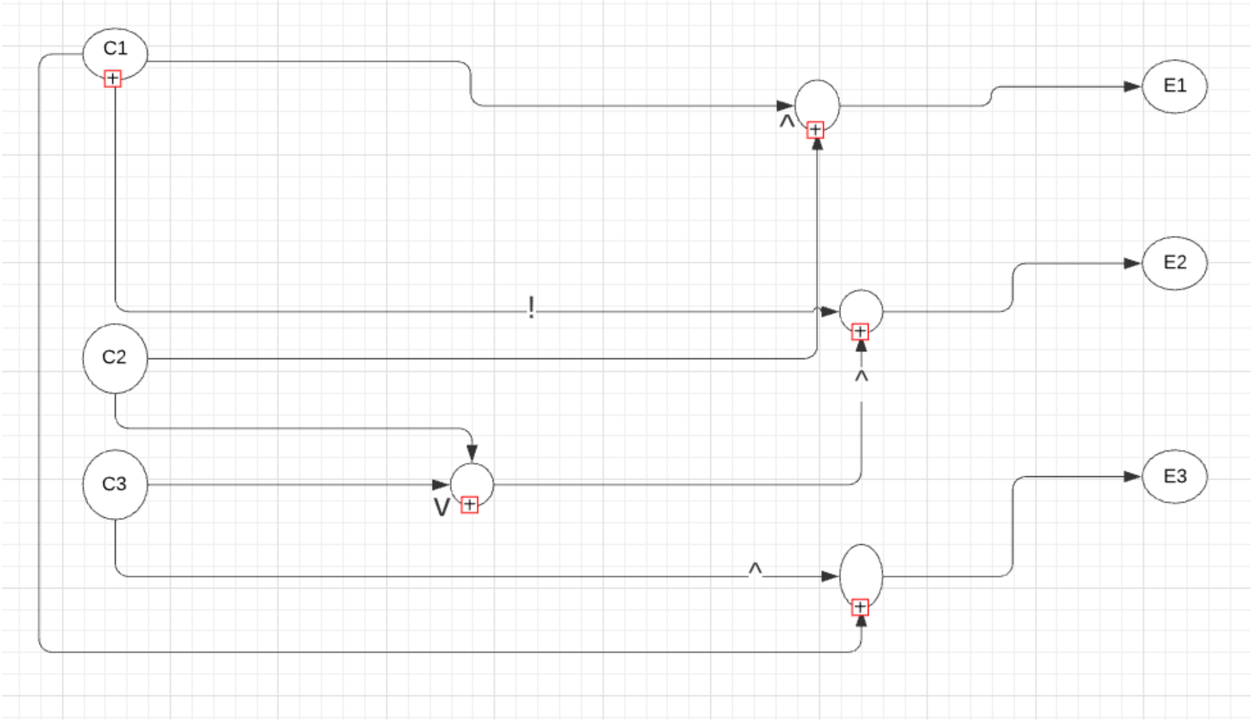
C3) Modification , accessing of the profiles.

Effects:

E1) Creation of profiles page is accessed by the manager.

E2) The system refuse the request of accessing the page.

E3) Modification , accessing and tracking the profiles page show up to the manager.



#	1	2	3
C1	1	0	1
C2	1	X	0
C3	0	X	1
E1	1	0	0
E2	0	1	0
E3	0	0	1

6.2) Test case specifications:-

6.2.1) Preconditions:

1) Manager was logged in to the system.

2) Profiles page was accessed by the manager for changing(creation ,accessing, modification).

3) Restaurant maximum number of staff is not reached yet.

On failure print “You are not eligible to make this operation”

6.2.2) Post conditions:

```
If (User is Manager)
Then
 {
While(MaxNumberOfStaff < Max_Restaurent)
If( Profilepage.change()==true)
Change in the profiles of the employees
}
Else
{
Print “Cant make this operation”
}
```

6.2.3) Operations:

1) Creation of the profiles of the new employees.

2) Accessing the profiles of the employees.

3) Modification in the employees profiles.

6.3) Test case:

Manager operations

Test Description(Scenario)	**Test data(Input)**	**Excepted results(output)**
Test the operation of creating the profiles of the employees by the manager.	1) User logged in as a manager. 2) If operation of creation of profiles is chosen. 3) Checking the maximum number of restaurant staff is not reached yet.	The manager created the profiles successfully, with the data is being updated on the database.
Test the operation of modification/accessing the profiles of the employees by the manager.	1) User logged in as a manager. 2) If operation of either modification/accessing of profiles is chosen. .	The manager can successfully accessing and modifying the profiles successfully.
Test of logging in to the system as any employee other than manager	1) User logged in as normal staff.	The system would print out “Cant access this page”

7) Authorize restricted waiter activities:

I will not model this feature as the SRS did not give much a detail about it and I will ask the QA department to ask the restaurant board who want the system for details about this feature.

8) Feature of tracking inventory and sales analysis.

8.1)The supportive models and assumptions for the extracted testable features:

Description of the features: The manager of the restaurant has the responsibility of tracking the inventory and the food goods, analyze the sales and profits of the restaurant and do other managerial operations.

Assumption of the feature:

1) I assumed at this feature that only the manager could do the operations of the restaurant and no other restaurant staff can do.

2) I assumed also that the manager could do his operations only by the free tab

Models: I will chose cause effect graph to model this feature**.**

Causes:

C1) Staff logged in as Manager Staff.

C2) the tab is free.

C3) Tracking the inventory operation.

C4) Sales analysis operation.

Effects:

E1) Tracking the inventory screen is available.

E2) Sales analysis screen is available.

E3) The system would be directed to a screen of unauthorized operation.

#	1	2	3	4
C1	1	1	0	X
C2	1	1	X	0
C3	1	0	X	X
C4	0	1	X	X
E1	**1**			
E2		**1**		
E4			**1**	**1**

8.2) Test case specifications:-

8.2.1) Preconditions:

1) User logged in successfully.

2) User account is a manager.

2) Data related to the restaurant is saved in the database for the needed analysis.

3) The inventory related data in the kitchen must be saved and accessed by the manager only.

On failure print “Cant perform this operation”

8.2.2) Post conditions:

```
If (User is Manager)
Then
 {
If(Data.Restaurent.available())
Perform sales analysis.
Else If( Data.Inventory().avaialble())
Track the inventory data
}
Else
{
Print “Cant perform this operation”
}
```

8.2.3) Variables/inputs:

1) Data of the restaurant.

2) The inventory to be tracked.

8.2.4) Operations:

1) Inventory tracking of the list of items and goods.

2) The sales analysis to calculate the profits for the restaurant.

8.3) Test case:

Tracking inventory and sales analysis

Test Description(Scenario)	**Test data(Input)**	**Excepted results(output)**
Test the operation of performing the sales analysis on the data of the restaurant stored in the database.	1) User is a manager. 2) Data is available to perform the analysis. 3) Data is stored in the database.	The manager can make the analysis operations to the restaurant.
Test the operation of tracking the inventory on the supporting goods and food packages of the restaurant stored in inventory page.	1) User is a manager. 2) Inventory of the packages is available for tracking.	The manager can track the inventory of the restaurant to see what packages is needed.
Test the system if the data is not available or there is a problem in database.	1) User logged in as a manager. 2) Database cant be accessed.	The system would print out "Cant perform this operation"
Test the system if the inventory is not available or isn't delivered yet.	1) User logged in as a manager. 2) Inventory of the packages is not available for tracking.	The system would print out "Cant perform this operation"

9) System testing:

- **Definition:** Is the type of testing in the software process, where it is done on a complete integrated system to ensure that the requirements are written specific to the system.
- **When it comes**: Usually it comes after the unit and integration testing (that detects any unusual outputs coming from the integrated tested units), where they served as an input to the system testing.
- **Purpose**: Is evaluation of the final user specification as to test the whole computer based system that have met the document and functionality requirements not only the software but also the other hardware components.
- **Type**: Black box testing.
- **Targets:** It targets the functional and nonfunctional requirements.

I will discuss the performance testing techniques.

9.1) Performance testing:

It is type of testing that targets the non functional requirements of a specific system (Restaurant management system) by checking its speed, reliability for the user view, so the goal of the performance testing is not discovering only the defects but for boosting the system performance and get confidence in front of the users and the market about the reliability and speed of the undergo system.

9.1.1) Load testing:

Definition: It is used for showing how the system would react under defined expected loads from the stakeholders.

As our system is not web based or mobile it targets only the restaurant staff and managers so the kind of requests and operations I mean here is for the local usage inside the restaurant as like in the features section and the requests with the other systems for tracking the inventory and others. So I will discuss the different loads on the system.

1. Waiter's tables assigned by host, changing their status (assume that the tables are 20).
2. Host notifying the busboy for cleaning the tables (assume that there are 3 terminals).
3. Waiter browsing the menu and taking the orders.
4. Waiter sending orders to the kitchen.
5. Manager do his managerial staff from create and accessing profile tracking inventory,etc..
6. Assuming our staff can be about 50 employees.

Note to be considered on all test cases all of them must be tested concurrently(working together) as our system need such a concurrent environment.

Test case #	**Test description(Scenario)**	**Test data(Input)**	**Expected result(output)**
1	Test logging in of staff employee to the system using the available terminals.	Test for about 25 employee Logging in.	I will see is the system login page would crashes or lags if more than 1 wants to login.
2	Test the assigning of many tables by the different hosts.	I will assign for about 10 different assigning's(normal workload) by the host.	I will see how the order page system would react to this load scenario or misuse the assigning of the tables for the correct waiters.
3	Test the waiter to take orders and select items and send it to the kitchen	Make the waiter to take 10 orders and sent it to the kitchen to view them .	Measure the system speed on the load and see whether it crashes or bottlenecks.

4	Test the tracking of the for the food and grocery packages, Analyze the sales and track the profiles of the employees.	Make enough requests from the manager to the outer systems	Measure the speed of sending these requests.

9.1.2) Stress testing

Definition : This type of testing ensure the stability and reliability of our system by forcing the system to heavy loads and see how the system works under it to give us the limit at which the system crashes beyond this point.

Test case #	**Test description(Scenario)**	**Test data(Input)**	**Expected result(output)**
1	I will test our software with abnormal activity/requests To see when the system would crashes.	1)I will assign for about 20 different assigning's(high workload) by the host. 2) Make the waiter to take 20 orders and sent it to the kitchen to view them. 3) Make the manager tracks the different food packages coming from other systems, and opens all the profiles of the staff. Note: This the maximum assigned requests for the software to handle technically and all tested concurrently.	See and reconfigure the output of the system when applying to this stress to see the 1) appropriate error message 2) The important data state(is it changed or not).

9.1.3) Soak and spike testing

Definition: Type of testing used to measure the performance and usage of the system under a tremendous load over some time where time here is an important factor to ensure the ability of the system to be under huge usage by the employee for long time.

Why used it? :-

1) As the restaurant is a large shift facility that can work for about 10 hours all in the 7 days so soak testing is an important kind of testing.
2) The restaurant also could have an abnormal conditions so it can lasts for 20 hours(for high spikes) as in holidays , in special occasions to see the system reaction at this cases as it is totally different that in the normal cases.

Test case #	**Test description(Scenario)**	**Test data(Input)**	**Expected result(output)**
1	I will use the above test cases by adding to it the time factor	I will keep using the system, and for an example I make a large number of log-ins to the system which will take a huge time.	1) I will see if there is a degradation of the system performance. 2) Is the database can outputs some errors or delete random profiles.

9.2) Recovery testing

Definition: Ensures that our system can successfully recover from any failure or crashes without any significant loss to data or the outgoing processes.

We can make it our system by having a single backup for the database or multiple(can be online or offline) that are already synchronized with the changes in the data for these types of failures.

Test case #	**Test description(Scenario)**	**Test data(Input)**	**Expected result(output)**
1	Cut the resources for building the system up,	1) Cut the network connectivity to outgoing process. 2) Deactivate parts of the system by the stress explained before.	1) See the results of the outgoing process with the other systems, and can the system restore it again. 2) See how these parts would activate again.

9.3) Configuration and compatibility testing

Definition: As our system is a computer based system , so we need to test not only the software but the other hardware and networks configuration.

Configuration important aspects:-

1) Software configuration: As our system would be on the terminals for host , tabs for waiter and other staff and screens for the kitchen so by choosing the appropriate language and frameworks we want to test our system what OS, what server platform and what database server should our system rely.
2) Hardware configuration: We should also test what devices our system use from printers, cameras, terminals, etc.

Compatibility important aspects:-

1) Is the system have on the tabs excel programs, data analytical tools and other software's for the managerial aspects and what version of office can the system works.
2) The other paper works of the system can work on what version of word, or on any document viewer.

Test case #	**Test description(Scenario)**	**Test data(Input)**	**Expected result(output)**
1	Wants to test the OS that the system works on	Test our system on Microsoft windows 7 and 10.	1) Check the compatibility of the system on the given OS.
2	Wants to see what hardware devices can be compatible for the output of the system	Test our system on various printers , security cameras and different technology.	1) Check and buy the hardware devices for the success of our system.

3)	Checks the system used software	1) Installation of different software editions and versions.	1) See what software and its version that provides our system with the best experience with least cost.

9.4) Usability testing:

Definition: Type of testing that reflects the easiness of use of the system by the users(staff) and here unlike most of the tests the user are actually use the system not the developers of it.

Test case #	**Test description(Scenario)**	**Test data(Input)**	**Expected result(output)**
1	Product wise usability testing.	Make outside clients try our system and other competing software's.	1) Judge our system and develop the uncompleted/lag Functionalities.
2	Is the software is user friendly	Can our staff of waiters, busboy and others actually use the system or it is a bit hard for them.	1) See our system can be used immediately after the deployment or needs training for the employees stuff.

9.5) Documentation testing:

Definition: Type of testing that tests the system functionalities, accuracy and bug tracking for the users who is not involved in the making of the system.

Test case #	Test description(Scenario)	Test data(Input)	Expected result(output)
1	Read test.	Test our documentation for the user to clarify its clarity, unambiguity and the flow of the documentation to describe the system.	1) Get the feedback from the users that actually read the document and then updates the document.
2	Hands-on test	Test our document as a user manual step by step from A-Z for a certain operation(like creating profile)	1) Get the feedback for the system manual of the features and updates the needed document.
3	Functional test	Actually, verify our document from the stated requirements with the client that needs the system.	1) Check to see if the document needs an update.

9.6) Acceptance testing:

Definition: Is type of testing that comes after the performance testing to determine whether or not our developed system has completed all the features and requirements with the end user and it is considered the last testing type before the deployment.

Types of acceptance testing:-

1) Pilot testing: We first employ our restaurant as small scale for one employee of each profession, to see the features can successfully operate on this controlled environment or not.
2) Alpha testing: Type of testing to discover and fix the bugs and it is marked as in-house testing and it can involves both the black box and white box testing, and from its characteristics that it is done early before the beta testing.
3) Beta testing: It is the last test before the deployment where a beta version of the software is released and to obtain the feedback on the system quality, is done by users who are not in the system development.

10) Security testing

10.1) Security architecture:

10.1.1) Definition:

Is the architecture of making a design that shows our system potential risks and what are the important assets that you primarily want to protect.

10.1.2) why is it important:

As known security is one of the most important aspects in any software as it can causes the data and information to be leaked. As the security architecture depends on subsystems, communication links and mechanisms position so before drawing the diagram I will first define each of them.

10.1.3) Subsystem:

10.1.3.1) Definition: Our architecture components that will provide or handle requests to the users.

7.1.3.2) Components:

1) Application system: Where our users can interact with and it is in the diagram of the architecture will be user interface layer or presentation layer.

2) Web server: It is the server that stores our site components file from normal (html files, images , JS file) where it is connected to the internet and handles the requests from our tabs/terminal and screens that are also connected to the internet and also the HTTP server which understands the URLs and the HTTP requests from our system to other external systems.

3) Application server: Is the server that run our system application and operation of the different features and processes the data coming and forwarding from the other servers.

4) DBMS server: Is the server that provides the services of the database to the restaurant tabs/terminals and screens by using the database application where the different systems provide this server with the needed queries.

10.1.4) Communication links

10.1.4.1) Definition: They are the links of communication that links between the different subsystems either internally between the different servers or externally with the internet and other systems.

10.1.4.2) Types of links:-

1) HTTPS: Secure version of normal HTTP (Hyper text transfer protocol)which is used to send the data between the browser of different restaurant hardware(tabs/terminals,..etc) and the website of the restaurant and used also when the patrons of the restaurant use the credit cards of the external banking systems which need to be secured.

2) SSL: (Secure security layer) is the security protocol for securing the links between different servers and browser server connection are encrypted.

3) TCP: (Transmission control protocol) Is the protocol that works with the IP for defining how the different programs can exchange data by setting up connection for the exchanging of data between the server and client.

10.1.5) Security mechanisms:

7.1.5.1) Definition: It is the mechanism that is used to find, prevent or recover from a security attack.

10.1.5.2) Types I will use::-

1) Authentication: Is the process of checking the identity of the person who will use the system, by verifying user credentials (As the staff in our restaurant, all of them will have username and password).

2) Authorization: Give the staff a higher privileges for accessing and using the resources(As the manager staff in our system will have an authorization higher than the other staff).

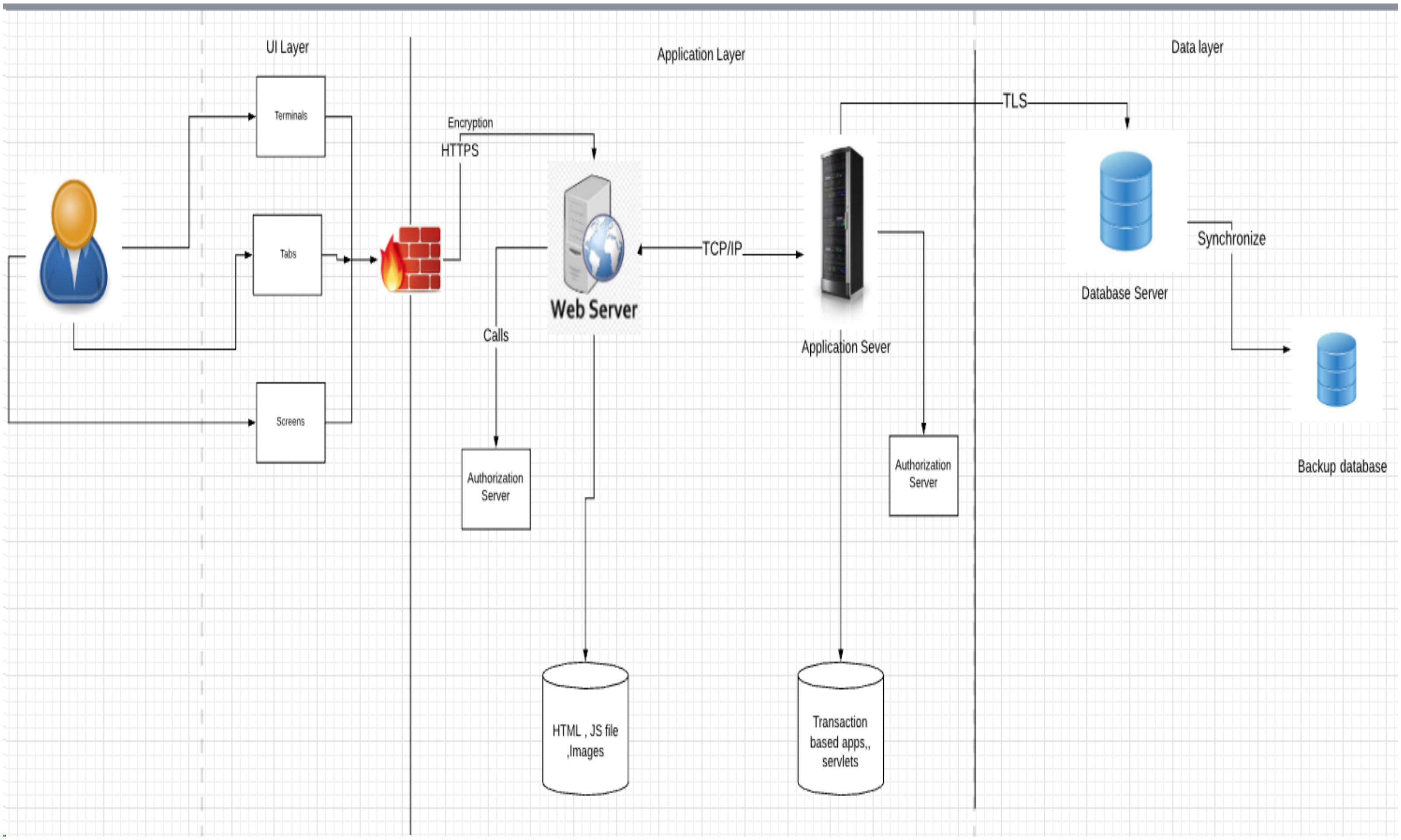
3) Encryption: The method of changing/cover the plain text by some way to hide its details for the attackers(All the passwords of my system staff login will be encrypted).

10.1.6) Security architecture:

I will divide my architecture into 3 layers/tiers, So I will use the 3 tier architecture where all layer contains their subsystems and each subsystem can call

Definition: 3-tier architecture is the architecture which composed of 3 layers which are:-

1) Separating the user experience in a **UI layer**
2) Business logic that consists of web and application server into **Application layer.**
3) **Data layer:** Where it contains the of database storage system and acess layer (MySQL,PostgreSQL,..etc) with a backup synchronized with our database in case of failure.



10.2) Security testing techniques:

Our QA team have planned a threat modeling process for discussing:-

1) The responsible team of experts and consultants will secure the restaurant system.
2) The important assets for protection:
    a. Credit cards of the patrons for confirming the order.
    b. Social security numbers of the employee staff.
    c. Our database server, which will have the valuable data for the restaurant. Payments, food inventory, their sales and profits.
    d. The domain server, which will costs them if they lost it.
3) Identify the possible threats for, viewing or changing data who will use the system legally/legitimately and who is unauthorized.
4) Document each threat what is their damaging potential, the corresponding financial loss, their discoverability level (is it easy to be discovered or not) and also the counter measures for each threat to follow to remove it.

So I will discuss the most important security techniques.

10.2.1) OS hardening

Is the test of making our OS securely, perform their regular updates and policies to help maintain our system in a fully protected mode by :-

a) Removing programs that is not very important as by limiting these programs will close a lot of options any attacker could use, By leaving the only important programs(Excel, Word ,…etc.) that are truly important for our restaurant.
b) Keep the system up-to-date with the latest versions and patches trusted from the house company.
c) Make the intended in charge technical persons ensure that the restaurant employees implements strong passwords and follow the security regulations.
d) Lock down the ports and use only them after checking its safety.

10.2.2) Penetration testing

Is the test with an intention for making a cyber-attack against our system for the sake of finding the security vulnerabilities/bugs that the attacker can use to violate the security by giving the attacker the possibility of entering and manipulating the API , servers.

From types of penetration testing is :-

a) Buffer overflow: The buffer is part of memory allocated for data of different datatypes of fixed size , and the overflow will occurs when we put data that the buffer cant handle.
b) Parameter tampering : Is the attack of playing with the parameters of the requests between the client and server to steal data or credentials.

10.2.3) SQL injection:

Is type of testing that prevents targeting the system that depends on databases heavily in which the attacker puts some SQL code to gain an access of some random important data. As for our system there is an important data for the employees, and the attacker could also run a random query that can either access or delete random or entire the database.

Prevented by:-

1) Avoid the statements/queries that allow any random user to input.
2) Use the encryption methods for the data.
3) Keep updating the databases.

10.2.4) Denial of service:

Is the test of forcing an excess of unusual traffic sent from a controlled host for the sake of shutting down the system making it unusual for normal/intended requests.

Symptoms on my system:

1) Degradation of performance.
2) Cant access its networks for long time.

10.2.5) IP spoofing:

Is the test that the prevents that attacker from tricking the users of the system they are communicating with the needed site but actually the attacker get access to the passwords or important data, Our restaurant system actually uses connected with the different systems for the inventory tracking feature.

Types of spoofing :-

1) IP address spoofing: Sending from false source address IP packets for the DOS attacks.
2) DNS server spoofing: Changing the path from the intended domain to different IP address.
3) ARP spoofing: Changing the mac address so that the our system can re directed to the malicious site/ attacker.

10.2.6) Password cracking

Is the test against the of breaking the users/ staff personal information by accessing the passwords, this maps in our feature of logging in the users to the system.

Vulnerabilities in passwords:-

1) Use default passwords: Either empty or very easy password.
2) Insecure password storing: To store the users important passwords unencrypted in easily found files.

11) References

1) Andreas Spillner, Tilo Linz, and Hans Schaefer. 2011. Software Testing Foundations: A Study Guide for the Certified Tester Exam (3rd. ed.). Rocky Nook.

2) https://www.lucidchart.com
Lucidchart is a web-based proprietary platform that is used to allow users to collaborate on drawing, revising and sharing charts and diagrams.