**System specification**

**Functionality specification**The functionality specification is closely related to the use cases. The main  
functions are ordering, \_\_\_ , \_\_\_, \_ , and administration. A description of each of  
the functionality will follow. It includes the output of the system and the input it  
expects from the users.

***Administration***includes adding, editing and deleting available ingredients, non-productive items, products and employees and viewing previous orders. The system will be able to show a list of ingredients and a list of products. For each ingredient, product a name, price and picture can be specified. The system also needs a function to show previous orders.

**Technical specification**The system is made up of three layers. At the top there is the GUI (Graphical User  
Interface) layer, the middle layer is the storage and query manager, the bottom layer is the underlying database.

***GUI layer***The GUI layer allows users to access the system. All the functionalities of the system must be available through the GUI. There are \_\_\_\_ separate GUI’s. One is for the customers to create orders and one is for employees for processing orders and administration purposes. With scripts, user input will be used to invoke queries from the storage and query manager layer to provide the user with various pages. The GUI should prevent input errors and in case of errors that could not be prevented, provide clear error messages.  
Therefore, customers do not have accounts to log on to.

*However customer information will be stored into the system to allow employees  
view previous orders by customers.*

The GUI for employees is on a separate URL. Users need a username and a password to gain access to the system. The administrator has access to add users to the system to give them access.

***Storage and query manager layer***The storage and query manager layer is responsible for information storage, retrieval, authorizations and error checking. This layer allows selecting, adding, updating, deleting entities and relations in the database by using different queries. Some instructions are limited to users with authorization, such as deleting data from the database which should only be allowed by the administrator.

***Database layer***The database layer contains all the data of entities and their relationships.

**1.4 Requirements**

**Functional requirements**

**Use cases:  
1. Customers**

**2. Employers/waiters**

**3. Administrators**

* The administrator must be able to log in and out.
* The administrator must be able to add/delete/edit orders.
* The administrator must be able to add/delete/edit ingredients.
* The administrator must be able to add/delete/edit un-useful products.
* The administrator must be able to add/delete/edit other users.
* The administrator must be able to view an order log.
* Only users with respective rights (administrators) must be able to use all these “Administrators” features.

**4.**

**5.**

**Database Design – Tables**

* Items – stores details of all items such as its name, price, category
* *Users* – stores details of all the users – customers, waiters/employers, and administers
* Oder – stores name of the order along with a timestamp and id of the

*lincked customer*

* Order\_details/ – stores the items corresponding to each order
* *Requests – stores all the Add/Delete update requests*
* *Pairs – stores which waiter is assigned to which table(customer)*
* Feedback/Comments – stores feedbacks corresponding to orders
* Gro/Menu – stores the categories of items ( )
* Customers –
* Employers/waiters –
* Administrators –
* Bill/Payment –

**5.**

**Non-functional requirements**

The system should be able to do the following constraints.

* Performance – system will give a response within a seconds
* Error Tolerance -