A

PROJECT REPORT

ON

“**Sales Management System**”

For

Final Semester Training

**Submitted by**

**Gagandeep Singh**

**1247671**

**CSE- 8A**

**ACKNOWLEDGEMENT**

I would like to take this opportunity to express my gratitude towards all the people who have in various ways, helped in the successful completion of my project.

I must convey my gratitude to ..................... for giving me the constant source of inspiration and help in preparing the project, personally correcting my work and providing encouragement throughout the project.I also thank all my faculty members for steering me through the tough as well as easy phases of the project in a result oriented manner with concern attention.

Thanking You,

**PROJECT OVERVIEW**

**1.1 THE PURPOSE OF THE PROJECT**

A case study at ‘Guckenheimer’ (an on-site corporate restaurant management and catering company) cited issues regarding a basic resources requirement list that has to be maintained manually by the staff. To keep track of their inventory levels they have to calculate a list of the groceries utilized during a course of time, calculate and analyze the requirements for the future, and place their next order to the vendors if needed. This process takes up a lot of time and human effort, and is also prone to human error. This poses a problem of a situation that the staff at ‘Guckenheimer,’ as well as many other restaurants faces. It takes up a lot of time to manually keep track of sales and place correct orders to vendors, wasting useful labor in trivial works. A product which would assist in tackling the above mentioned problems would prove to be fruitful to clients such as ‘Guckenheimer’ and similar enterprises as this product would help convert the unproductive time to something more useful, by removing the unnecessary error prone complications and efforts.

**1.2 GOALS OF THE PROJECT**

The project aims at providing an efficient interface to the restaurants for managing their grocery inventory based on each item sold. The basic idea involved here is that each item is linked to its atomic ingredients which are stored in a database. At the end of each day, the system analyzes the total sale of menu items and proportionately deducts appropriate amount from the resource database. Then it compares the current available resources with the threshold level of each ingredient. If it finds that certain ingredients are below the threshold, it will generate a purchase order for those item(s) and send it to the manager (admin) for approval. We also propose to include a special feature “Prediction”. This feature keeps track of any upcoming occasions, climatic changes and special events that may influence inventory needs for the upcoming week. The system will then predict the required resources for these events based on previously accumulated information/knowledge. It will now generate an updated purchase order in accordance with the predictions. The product also aims to keep track of the shelf life of resources. If any resource nears the end of its shelf life, it would intimate to the manager (admin) the details of the quantity that is near its expiration date. The restaurant must function efficiently, the groceries must be tracked correctly, timely orders must be sent out to the vendors, and the inventory must be maintained and updated at all times.

**1.3 THE DOMAIN**

This proposed project aims at inventory control in the restaurant and catering Industry. Such a large domain would result in an equally as large scope of development. As a result we narrow our software down to our case study of an outlet of Guckenheimer concentrating only on the basic resources utilized in inventory control of the outlet. Although the software will be developed keeping in mind the needs of 10 Guckenheimer and available data at first, then applying it to the larger domain of the entire restaurant industry can be achieved with ease. Our target domain is full of software to track sales of food items, but lacks in this area of inventory management. Our software can be scaled from large corporate dining all the way to small privately-owned restaurants. It is also fairly domain specific: the database runs off recipes which generate the necessary ingredients. It also updates the inventory based off of the sale of those recipes. This requirement focuses our product to our domain and makes it more appealing to those looking for a solution to this specific problem.

**1.4 THE CLIENT**

The client can vary from private restaurant owners to corporate restaurant management companies, such as Guckenheimer (www.guckenheimer.com). A corporate restaurant management company that starts up, staffs, and oversees the everyday workings of a corporate restaurant, such as the one in the Groupon Chicago office. As stated above, while our product can be applied to the entire domain of the restaurant and catering business, focusing on a specific business provides us with more precise and consistent data. A company such as Guckenheimer would be an ideal client, as they staff multiple corporate kitchens across the nation. A large scale company such as this this can apply our software to each and every kitchen, cutting down costs on a very large scale. Our software will allow our client to customize the database to suit the needs of each kitchen individually. They can vary in recipes, vendors from which they order their products, and threshold levels. This provides a uniform product that can be customized at a smaller scale. Our client would need to purchase multiple licenses, or more likely a corporate subscription that would allow them to use the software in multiple kitchens. We would also offer single use licenses to appeal to restaurants that only need to manage a single inventory of goods.

**1.5 USER OF THE PRODUCT**

The main users of the product would be kitchen management and staff. The management would approve the orders that would be sent out, provide vendor information, upload recipes, and set threshold levels. Many of these tasks, such as the information regarding vendors, recipes, and threshold levels would need to be set only once. Of course, the option to add, remove, or update this data would be implemented as well. Once this initial step has been taken, our software will require nothing more than a weekly approval for the orders being sent out, minimizing the work that management has to complete in order to insure the correct amount of inventory is available. Kitchen staff would be responsible for updating the amount of product sold at the end of the day. Each day, the register prints out the products sold and the quantity of each product sold. Instead of manually subtracting that amount from the inventory, they input the amounts sold into our software which will do the number crunching for them. This data is also stored into the “predictions” feature for future use. 11

**1.6 OBJECTIVES AND SUCCESS CRITERIA OF THE PROJECT**

The objective of the project is to provide an efficient inventory control whose main functionality apart from calculating the inventory include predicting the requirement for the next order and also if there is a “Special Occasion” then accordingly the manager selects the particular occasion and extra requirements is added to the next issuing order to the vendors which needs to be approved by the manager. The product also aims to keep track of the shelf life of resources. If any resource nears the end of its shelf life, it would intimate to the manager (admin) the details of the quantity that is near its expiration date. The success criteria depends on

 The accuracy in maintaining the inventory levels

 The accuracy in predicting the requirements of the next order

 The accuracy in relating recipes to their respective ingredients

 Ease of use when it comes to updating inventory levels and placing orders to vendors

**2 REQUIREMENTS ANALYSIS**

**3.1 FUNCTIONAL REQUIREMENTS**

 The user must have, at disposal, functions for managing the inventory efficiently.

 The functions for inventory management should allow the user to know which ingredients in the inventory are below their threshold levels and need attention.

 The system must include functions that will allow the user to add a recipe, ingredient, vendor to the database.

 The user should also be able to delete any recipe from the database when not needed.

 The system must allow the user to create orders for the ingredients that are below threshold.

 The system must include a mechanism for the user so that the user can just update the sales of the day in the system and the system deducts the corresponding amount of ingredient quantity from the inventory. Thus keeping a track of ingredients.

 The system must also include functions for the user to add special days in the system when the inventory usage will be more than usual or less than usual and thus provide a way to alert the user of the possibility of over usage or under usage or certain ingredients.

 The system also must provide a prediction function to the user where the system will give the user the predicted usage of inventory of certain pre-set days.

 The system must have a password protected access system such that only people with authenticated credential are allowed to access the function of the system.

**3.2 NON-FUNCTIONAL REQUIREMENTS**

** Usability**

i. The system must be easy to use by both managers and chefs such that they do not need to read an extensive amount of manuals.

ii. The system must be quickly accessible by both managers and chefs.

iii. The system must be intuitive and simple in the way it displays all relevant data and relationships. iv. The menus of the system must be easily navigable by the users with buttons that are easy to understand. \*\*Note: For detailed Requirements, refer to the Requirement Analysis Document [9]. 15

** Reliability**

i. The System must give accurate inventory status to the user continuously. Any inaccuracies are taken care by the regular confirming of the actual levels with the levels displayed in the system.

ii. The System must successfully add any recipe, ingredients, vendors or special occasions given by the user and provide estimations and inventory status in relevance with the newly updated entities.

iii. The system must provide a password enabled login to the user to avoid any foreign entity changing the data in the system.

iv. The system should provide the user updates on completion of requested processes and if the requested processes fail, it should provide the user the reason for the failure.

v. The system should not update the data in any database for any failed processes.

** Performance**

i. The system must not lag, because the workers using it don’t have down-time to wait for it to complete an action.

ii. The system must complete updating the databases, adding of recipe, ingredient, vendor and occasions successfully every time the user requests such a process.

iii. All the functions of the system must be available to the user every time the system is turned on.

iv. The calculations performed by the system must comply according to the norms set by the user and should not vary unless explicitly changed by the user.

** Supportability**

i. The software is designed such that it works even on systems having the minimum configuration.

ii. The system is adaptable even if additional plugins or modules are added at a later point.

iii. The data can be exported to the manager so as to make the system more portable. 16

** Packaging**

i. The system must be able to run on the Windows operating systems beginning with Windows XP, and must be able to run on future releases such as the upcoming Windows 8

ii. The software must incorporate a license key authentication process.

iii. The packaging must come with a manual that details the use of the system, and also the instructions on how to use the program. This manual may be included either in a booklet that comes with the software, or on the disc that the software itself is on.

**3 DETAILED SYSTEM DESIGN**

**3.1 DESIGN GOALS**

** Low Response Time:**

The main functionality of the system involves updating and reading the data from the database for different entities such as ingredients, recipes vendor etc. Thus the time required to retrieve/ update/ add data to the database should be minimum and preferably should be in the range of 2-5 seconds or lesser.

** High Robustness:**

The system should constantly check the user input at all instances that could generate errors in the program. For instance: o The system should be able to check input values for the amount of ingredients required for the recipe and should make sure the user enters a numeric value in the input box and the system shows an error and asks the user to re-input if in a perfectly validated field an improper data type is inputted. o The System should have validated input data fields and must put a constraint on the inputted names of recipe, ingredient, vendor, and occasion etc. to ensure no duplicate entries are added in the database. This ensures the robustness of the maintained database. o The system should verify all the inputs by the user by using a confirmation dialog box before processing and making changes to the data.

** High Reliability:**

The reliability of the system depends upon its ability to replicate the specified behavior. The safekeeping of the data is essential so as a result a backup of the levels is generated and stored in the warehouse. There are numerous factors on which reliability can be defined as for example, the specifications mention that the updating of database or the notification of a successful update must be carried out within 2-5 seconds of initiation and the system must adhere to these specifications to be called a reliable system. Similarly, the system should be able to achieve performance in lieu with the specifications mentioned.

** Low fault tolerance:**

The system works on sensitive data and therefore any fault in the functioning of the system will hinder accurate updating or reading of data. This could lead to invalid entries in the database. Thus, the system should have low fault tolerance. This is in tandem with the design goal of high robustness as the validation checks to ensure correct inputs from the user implies that the fault tolerance of the system is low. \*\*Note: For detailed Design Report, refer the Design Report submitted earlier. [10] 38

** Security:**

The system must provide a login functionality to the Manager as the manager is the authenticated controller of the system and any other user is not permitted to use the system functionality and make changes in the database. Thus proper user authentication should be necessary before system launch.

** High Extensibility:**

The design of the system should be such that any future improvement can be added with no or minimum improvements. It is in one’s best interest to always give space for future enhancements. For instance, right now there is no class that will help the user to manipulate prediction of values and the current system only predicts the ingredient usage for a certain date, but a feature can easily be added to incorporate prediction of recipe usage, order prediction etc.

** Low Adaptability:**

The system is designed to work on the domain of inventory control and management in the restaurant and catering industry. The functioning of this project is limited only to these particular businesses which have similar functioning and thus it would then be subject to structural re-modification in order to to apply it in some other application domain.

** High Readability:** The system code should be properly commented so as to explain the functionality of the code fragments. The code comment should explain the function or task the code fragment performs and the result and the return value of the corresponding function or task should also be mentioned.

** High Traceability:** The coding scheme of the system should be such that it could be traced back to its requirements specifications. This will enable high traceability of the code of the system.