**Software Requirement Specification (SRS)**

**Food Safety Portal**

**1. Introduction**

**1.1 Purpose**

Food Safety Portalis a web application which provides information related to restaurants of a city, market certifications,quality of food, dining guide, food handling, local diet/delicacies to everyone at their doorstep.

The users living in the city or visitors of that city can connect through internet can get information about restaurants of that city, food related illness and restaurant guide.

City employees can easily update an information and others users also can submit feedback, suggestion for improvement on the web-site.

**1.2 Scope**

* There are three basic users - city residents, visitors and city employees.
* All users have their own profiles in FSP.
* City residents and visitors can search for restaurants and make online booking. They can also submit feedback on any restaurant.
* City employees can view record of their customers (regular customers, weekend customer or new customers).
* Restaurant guide views and manages the booking, performs day open and close activitiesand calculates his commission. He also sends reports to admin.
* Admin has the authority to add/delete users, grant permission to city employees and restaurant guide to generate and view. He also views complainsand feedback of users and takes necessaryactions.

**1.3 Definitions, Acronyms and Abbreviations**

**Acronyms and Abbreviations:**

FSP : Food Safety Portal

Admin : Administrator

RG : Restaurant Guide

WASCE : Web Sphere Application Server Community Edition

DB2 : Database\_2

JSP : Java Server Pages

J2EE :Java 2 Enterprise Edition

UML : Unified Modeling Language

XML : Extensible Markup Language

HTTP : Hypertext Transfer Protocol

RAD : Rational Application Developer

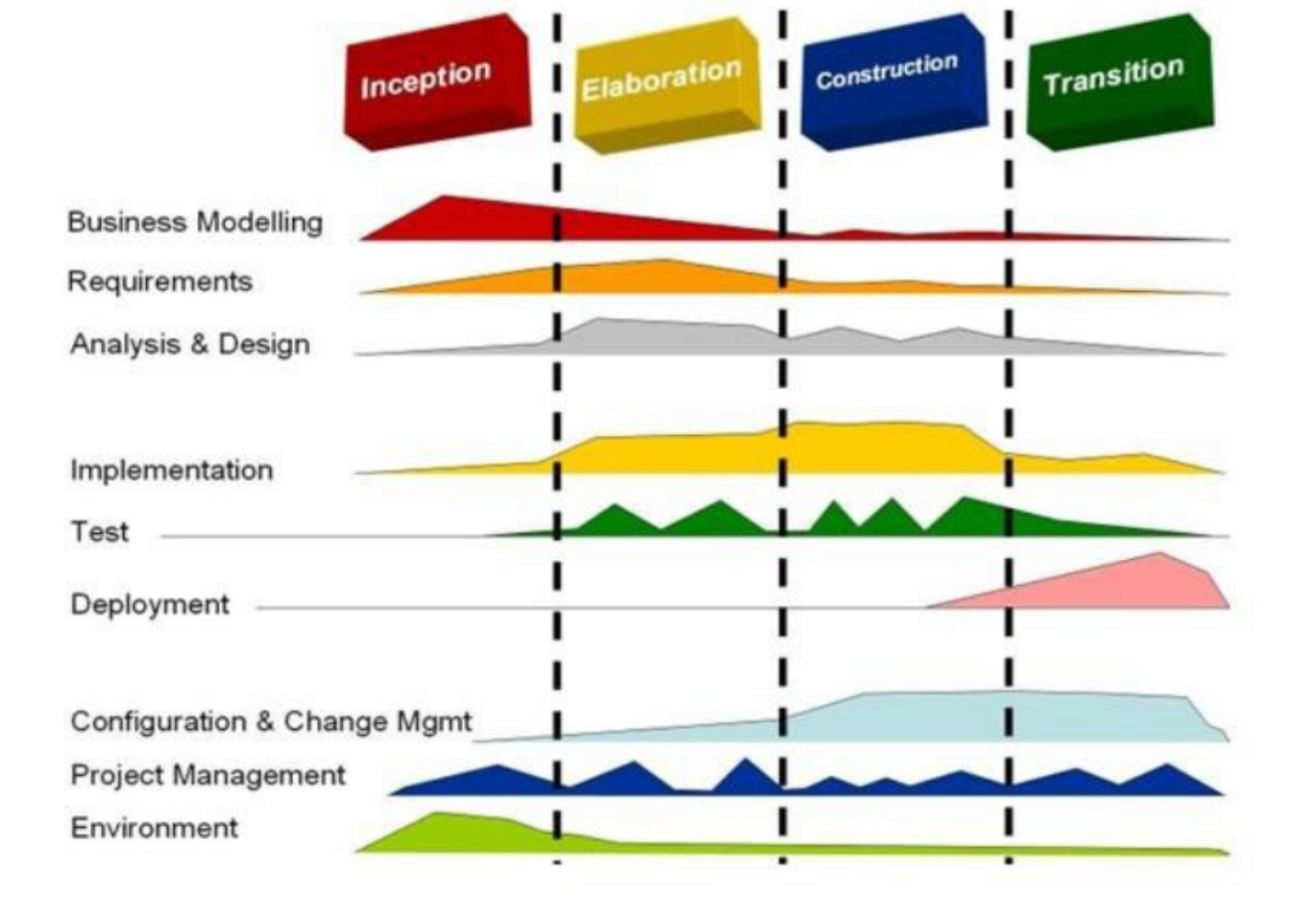
**Definitions:**

**Inception:**The goal of the inception phase is to establish a business case for the system. Identifying all external entities that will interact with the system and defining these interactions. This information is used to assess the contribution of system to business.

**Elaboration:**The goals of the elaboration phase are to develop an understanding of the problem domain, establish an architectural framework, develop project plan and identify key project risks.

**Construction:**This phase is concerned with system design, programming and testing. Parts of the system are developed in parallel and integrated during this phase.

**Transition**:This is the final phase of RUP and is concerned with moving the system from the development community to the user community and making it work in real environment.



**1.4 References**

* Object Oriented Modeling and Design with UML-Michael Blaha, James Rambaugh.
* Software Engineering, Seventh Edition, Ian Somerville.
* IBM Red Books.
* IBM TGMC Sample Synopsis.
* IBM – www.ibm.in/developerworks.
* Java - www.sun.com
* Wikipedia - *www.wikipedia.com*
* Database Management Systems – Navathe
* Complete Reference - J2EE - Keogh.
* Software engineering concepts and practices by Ugrasen suman.

**1.5 Overview**

**Existing System:**

* Registration for users
* Discussion forum

**Drawbacks:**

Remote area users and people who doesn't have knowledge of internet cannot use the system.

**Proposed System:**

Registration for city residents, visitors, city employee - help the users who doesn't have knowledge of internet to use the system.

**Our Plan:**

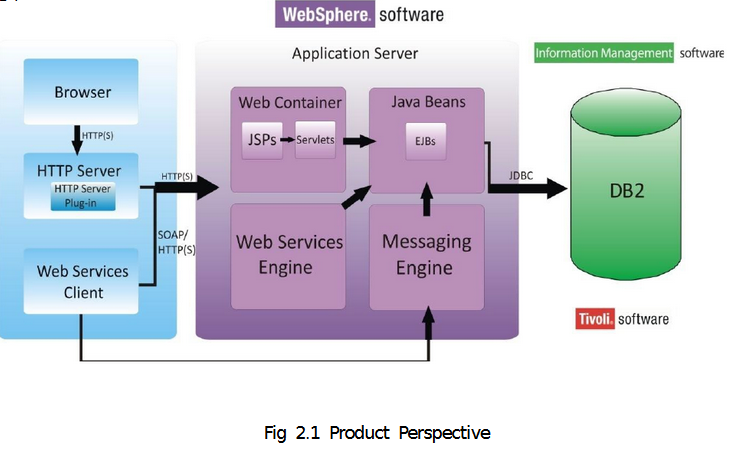
* Registration for users.
* Online maintenance of restaurant record.
* Online booking facility.
* Users reviews and ratings for restaurants*.*

**2. General Description**

**2.1 Product Perspective**

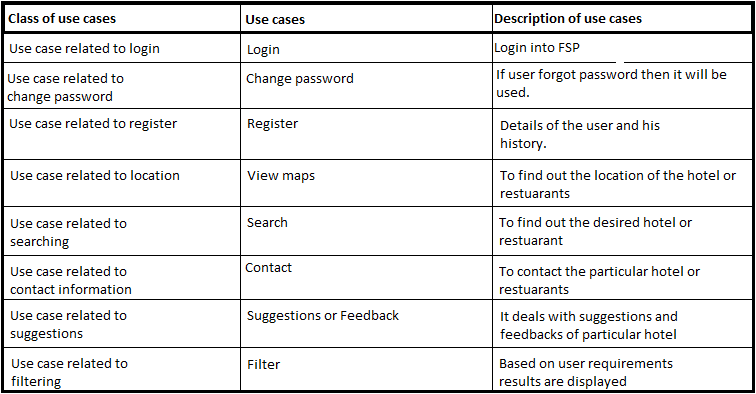
FSP is aimed toward the people who considers about their health and to find healthy food in restaurants and hotels where it provide an application which allows residents and users to find out more about food-related aspects about a particular city. FSP should be user-friendly, ‘quick to learn’ and reliable software for the above purpose.

FSP is intended to be a stand-alone product and should not depend on the availability of other software. It should run on both UNIX and Windows based platform.

****

**2.2 Product Functions**

FSP should support the following use cases:



**2.3 User characteristics**

1.The user should be familiar with the Food safety portal and google maps.

2. User should be computer literate.

**2.4 Principal Actors**

The two principal actors in FSP are “user” and “system”.

**2.5 General Constraints**

a. For full working FSP requires Internet connection.

b. FSP is single-user application

**2.6 Assumptions and Dependencies**

a. Full working of FSP is dependent on the availability of Internet connection.

b. The foods items are displayed based on limitations of restaurants and hotels

**3. Specific Requirements**

**3.1 Functional Requirements**

* Web accessible information base
* Provide templates for information entry – e.g. education, dining guide, food handling guide, etc.
* Allow for easy update of information by city employees
* Allow for easy retrieval of feedback collected to facilitate acting on feedback received
* Extensible to allow each city to update with their own specific information
* Allow report of food-related illness (non-life-threatening)
* Allow report of food safety concern
* Allow submission of suggestions for improvement.
* Enable amap view of the city which marks locations of the restaraunts and markets.

**3.1.1 Use cases of the Functional Requirements**

We describe the functional requirements by giving various use cases.

Use cases related to system authorization:

**Use Case 1:** Register

Primary Actor: User

Pre Condition: Nil

**Main Scenario:**

1. Start the application. User prompted for regiser

2. User gives the details.

3. System does stores information.

4. Main screen is displayed with confirmation email.

**Alternate Scenario**:

1. Authorization fails

2. Prompt the user that he typed the wrong credentials

Use cases related to system authorization:

**Use Case 2:** Login

Primary Actor: User

Pre Condition: Nil

**Main Scenario:**

1. Start the application. User prompted for login and password.

2. User gives the login and password.

3. System does authentication. .

**Alternate Scenario**:

1. Authorization fails

2. Prompt the user that he typed the wrong password

3. Allow him to re-enter the password. Give him 3 chances.

**Use Case 3:** Change Password

Primary Actor: User

Pre Condition: User logged in

**Main Scenario:**

1. User initiates the password change command.

2. User is prompted for old password, new password and confirm new password.

3. User gives the old password, new password and confirm new password.

4. System does authentication.

5. New password is registered with the system.

**Alternate Scenario**:

1. Authorization fails.

2.Prompt the user that he typed the wrong password.

3. Allow him to re-enter the password. Give him 3 chances.

4. New password and confirm new password do not match.

5.Allow him to re-enter the attributes. Give 3 chances.

Use cases related to USERS:

**Use Case 4**:view maps

User Pre Condition: User logged in.

**Main Scenario**:

1. User views the google maps

2. System asks the user for the area to viewed.

3. User enters the area name and the hotel name.

4. Maps related to that area and hotel is displayed is displayed.

**Alternate Scenario:**

1.No internet connection.

2.No results found related to that data entered.

3.Invalid area name or hotel name .

**Use Case 5** :search Hotels or restaurants.

Primary Actor: User

Pre-Condition: User logged in.

**Main Scenario**:

1. User searches the hotels and restaurants.

2. System displays the list of hotels or restaurant available.

3. User enters the hotels name.

4. Information regarding the Hotels foods, address, delivery options etc .. displayed.

**Alternate Scenario:**

1.Only limited number of hotels are available in case of new hotel results in no information.

**Use Case 6**: Contact Hotel or Restaurant and book deal

Primary Actor: User

Pre-Condition: User logged in.

**Main Scenario**:

1. User searches the hotels and restaurants contacts .

2. System displays the list of hotels or restaurant contacts available.

3. User enters the hotels name .

4. Information regarding the Hotel management information ,address, delivery options etc .. displayed.

**Alternate Scenario**:

1.Only limited number of hotels are available in case of new hotel results in no information.

2.If hotel has no contact information then no results are displayed

**Use Case 7** :Search Hotels or restaurants quality or feedback.

Primary Actor: User

Pre-Condition: User logged in.

**Main Scenario:**

1. User searches the hotels and restaurants.

2. System displays the list of hotels or restaurant available.

3. User enters the hotels name.

4. Information Bulletins regarding the Hotels Quality, taste, ratings, market certifications foods, address, delivery options etc displayed.

**Alternate Scenario:**

1.Only limited number of hotels are available in case of new hotel results in no information.

2.For some Hotels and restaurants there may no ratings,quality,feedback given by market officials.

**Use Case 8**: search Hotels or restaurants status and certifications

Primary Actor: User

Pre-Condition: User logged in.

**Main Scenario:**

1. User searches the hotels and restaurants .

2. System displays the list of hotels or restaurant available.

3. User enters the hotels name.

4. Information Bulletins regarding the Hotels status either 5star or 4 star or 3 star displayed followed by market certifications given by government.

**Alternate Scenario**:

1.Only limited number of hotels are available in case of new hotel results in no information.

2.For some Hotels and restaurants there may no certifications given by market officials.

**Use Case 9**: user suggestions,rating,feedback

Primary Actor: User

Pre-Condition: User logged in.

**Main Scenario** :

1. User logins and gives comments or feedbacks to the website management .

2.User can also give suggestions to the particular hotel visited and can give ratings also.

3.System must show user information regarding the past reviews and display later when he again user use it..

4.System must store the reviews given by the user and display the thanks alert message on the user interface.

**Alternate Scenario:**

1.Only limited number of hotels are available in case of new hotel results in no information.

2. Some Hotels and restaurants are not yet registered in such cases it is not possible to give feedback and results in error.

**Use Case 10:** Update complaints status

Primary Actor: City employee

Pre Condition: Login

**Main Scenario**:

1. City employee screen is displayed.

2. The city employee updates the status of the complaint in reference to the progress of it.

3. The complaints which are solved are deleted.

**Alternate Scenario**:

4. No updatations are present.

**Use Case 10:** View new complaint

Primary Actor: City employee

Pre Condition: Login

**Main Scenario:**

1. City employee screen is displayed.

2. The city employee opens the new complaint menu.

3. Forwards the new complaints to higher authorities.

Alternate Scenario:

4. No new are present.

**Use Case 11: Give solution**

Primary Actor: City employee

Pre Condition: Login

**Main Scenario**:

1. City employee screen is displayed.

2. The city employee gives solution to the complaints sent by the city residence.

3. The solutions are given by the higher authorities.

**Alternate Scenario**:

4. Post message to the user screen that solution is still pending.

**Use Case 12:** Forwardcomplaints

Primary Actor: City employee

Pre Condition: Login

**Main Scenario**:

1. City employee screen is displayed.

2. The city employee forwards the complaint to the supervisor to take action on the complaint.

3. Posts a message to the users that their complaint was forwarded to higher authorities.

Use case related to Residence:

**Use Case 13:** Update profile

Primary Actor: City resident

Pre Condition: Login

**Main Scenario**:

1. City resident screen is displayed.

2. The city resident update their profile which contain the data originally submitted by them to the portal.

**Alternate Scenario**:

3.Every information is up to date.

**Use Case 14:** Give complaint

Primary Actor: City resident

Pre Condition: Login

**Main Scenario**:

1. City resident screen is displayed.

2. The city resident give complaints to the portal using the templates form respectively.

3.The resident can give complaint to any of the authority.

Alternate Scenario:

4.No complaints present in the city in city residents view.

**Use Case 15:** Check complaint status

Primary Actor: City resident

Pre Condition: Login

**Main Scenario**:

1. City resident screen is displayed.

2. The city resident check the compliant status updated by the city employee.

**Alternate Scenario**:

3. Repost the complaint to the authority saying that his posting the complaint for the second time because of the zero status.

**Use Case 16:** Feed back

Primary Actor: City resident

Pre Condition: Login

**Main Scenario**:

1. City resident screen is displayed.

2.The city resident gives the feedback regarding the services provided by the city employee and also regarding the status of their complaint.

3. He also chooses one checkbox out of good, better and average.

**3.2 External Interface Requirements**

**3.2.1 GUI**

The Software provides good graphical interface for the user and the administrator can operate on the system. Performing the required task such as create an account, services, about us of portal.

* It allows user to view information.
* The user interface must be customized by only the administrator.
* All the modules provided with the software must fit into the graphical user interface and accomplish to standard defined.
* The design should be simple and all the different interfaces should follow a standard template.
* The user interface should be able to interact with the user management module and a part of the interface must be dedicated to the login/log out module.

**Login Interface**:-

In case the user is not yet registered, he can enter the details and register to create an account.

Once his account is created he can ‘Login’ which asks the user to type his Username and password. If the user entered either his username or password incorrectly then an error message appears.

**Categories View**:-

Categories view shows the categories of area, city and place and provides the administrator to add/edit or delete category from list.

**Control panel**:-

This panel will allow the administrator to add/remove users. And manage lending options.

**3.3 Performance Requirements**

The proposed system that we are going to develop will be used as the Chief performance system throughout the area, city for all the users. Therefore, it is expected that the database would perform functionally all the requirements that are specified by the users.

* The performance of the system should be fast and accurate.
* Food safety portal shall handle all the expected and un-expected errors in the ways that prevent the loss of information and long down-time period. Thus it should have inbuilt error testing to identify invalid username/password.
* The system should be able to handle large amount of data. Thus it should accommodate high details without any fault.
  1. **Design constraints**

Design Engineers must consider a multitude of technical, economic, social and environmental constraints when design products and processes. There must be clear evidence in your design project that you have addressed constraints that are relevant to your project.Specify design constraints that can be imposed by other standards, hardware limitations, etc*.*

**3.5 Security requirements**

Security systems need database storage just like many other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully.

Specify the factors that would protect the software from accidental or malicious access, use, modification, destruction, or disclosure. Specific requirements in this area could include the need to:

1. Utilize certain cryptographic techniques
2. Keep specific log or history data sets
3. Assign certain functions to different modules
4. Restrict communications between some areas of the program
5. Check data integrity for critical variables

Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.

**3.6 Maintainability Requirements**

Specify attributes of software that relate to the ease of maintenance of the software itself. There may be some requirement for certain modularity, interfaces, complexity, etc. Requirements should not be placed here just because they are thought to be good design practices. If someone else will maintain the system.

Specify attributes of software that relate to the ease of porting the software to other host machines and/or operating systems. This may include:

1. Percentage of components with host-dependent code
2. Percentage of code that is host dependent
3. Use of a proven portable language
4. Use of a particular compiler or language subset
5. Use of a particular operating system

**3.7 Reliability Requirements**

Specify the factors required to establish the required reliability of the software system at time of delivery. If you have MTBF requirements, express them here. This doesn’t refer to just having a program that does not crash. This has a specific engineering meaning.

**3.8 Availability Requirements**

Specify the factors required to guarantee a defined availability level for the entire system such as checkpoint, recovery, and restart. This is somewhat related to reliability. Some systems run only infrequently on-demand (like MS Word). Some systems have to run 24/7 (like an e-commerce web site). The required availability will greatly impact the design. What are the requirements for system recovery from a failure? “The system shall allow users to restart the application after failure with the loss of at most 12 characters of input”.

**3.9 Database requirements**

The basic objective of normalization is to be reduce redundancy which means that information is to be stored only once. Storing information several times leads to wastage of storage space and increase in the total size of the data stored.

If a Database is not properly designed it can gives rise to modification anomalies. Modification anomalies arise when data is added to, changed or deleted from a database table. Similarly, in traditional databases as well as improperly designed relational databases, data redundancy can be a problem. These can be eliminated by normalizing a database.

Normalization is the process of breaking down a table into smaller tables. So that each table deals with a single theme. There are three different kinds of modifications of anomalies and formulated the first, second and third normal forms (3NF) is considered sufficient for most practical purposes. It should be considered only after a thorough analysis and complete understanding of its implications.

3.10 Specific Requirements

This section contains all the software requirements at a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements. Throughout this section, every stated requirement should be externally perceivable by users, operators, or other external systems. These requirements should include at a minimum a description of every input (stimulus) into the system, every output (response) from the system and all functions performed by the system in response to an input or in support of an output. The following principles apply.

1. Specific requirements should be stated with all the characteristics of a good SRS
2. correct
3. unambiguous
4. complete
5. consistent
6. ranked for importance and/or stability
7. verifiable
8. modifiable
9. traceable
10. Specific requirements should be cross-referenced to earlier documents that relate
11. All requirements should be uniquely identifiable (usually via numbering like 3.1.2)
12. Careful attention should be given to organizing the requirements to maximize readability (Several alternative organizations are given at end of document)

**3.11 Safety Requirements**

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.

**3.12 Operation Requirements**

Specify the normal and special operations required by the user such as:

1. The various modes of operations in the user organization
2. Periods of interactive operations and periods of unattended operations
3. Data processing support functions
4. Backup and recovery operations

This is sometimes specified as part of the User Interfaces section. If you separate this from the UI stuff earlier, then cover business process type stuff that would impact the design. For instance, if the company brings all their systems down at midnight for data backup that might impact the design. These are all the work tasks that impact the design of an application, but which might not be located in software.

**3.13 Site adaptation Requirements**

In this section:

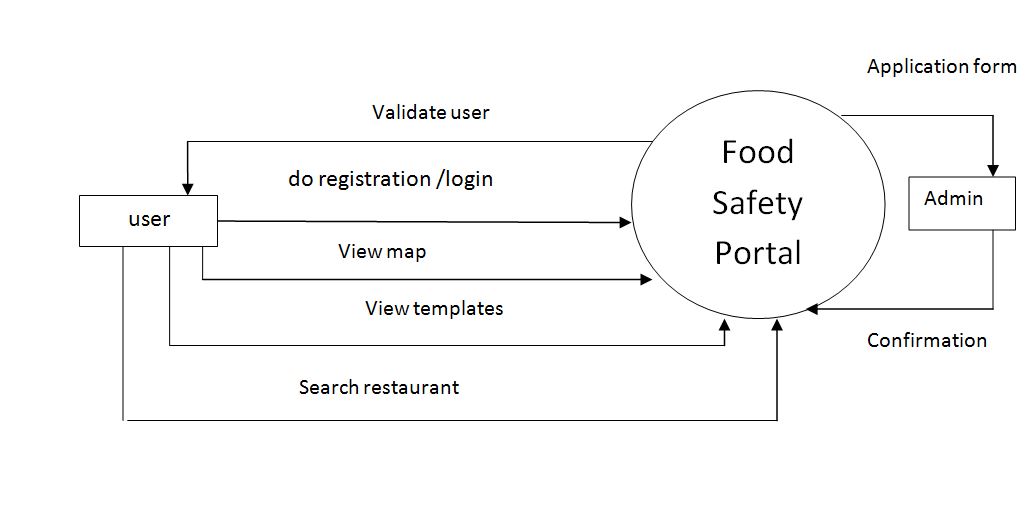
1. Define the requirements for any data or initialization sequences that are specific to a given site, mission, or operational mode
2. Specify the site or mission-related features that should be modified to adapt the software to a particular installation

If any modifications to the customer’s work area would be required by your system, then document that here. For instance, “A 100Kw backup generator and 10000 BTU air conditioning system must be installed at the user site prior to software installation”.

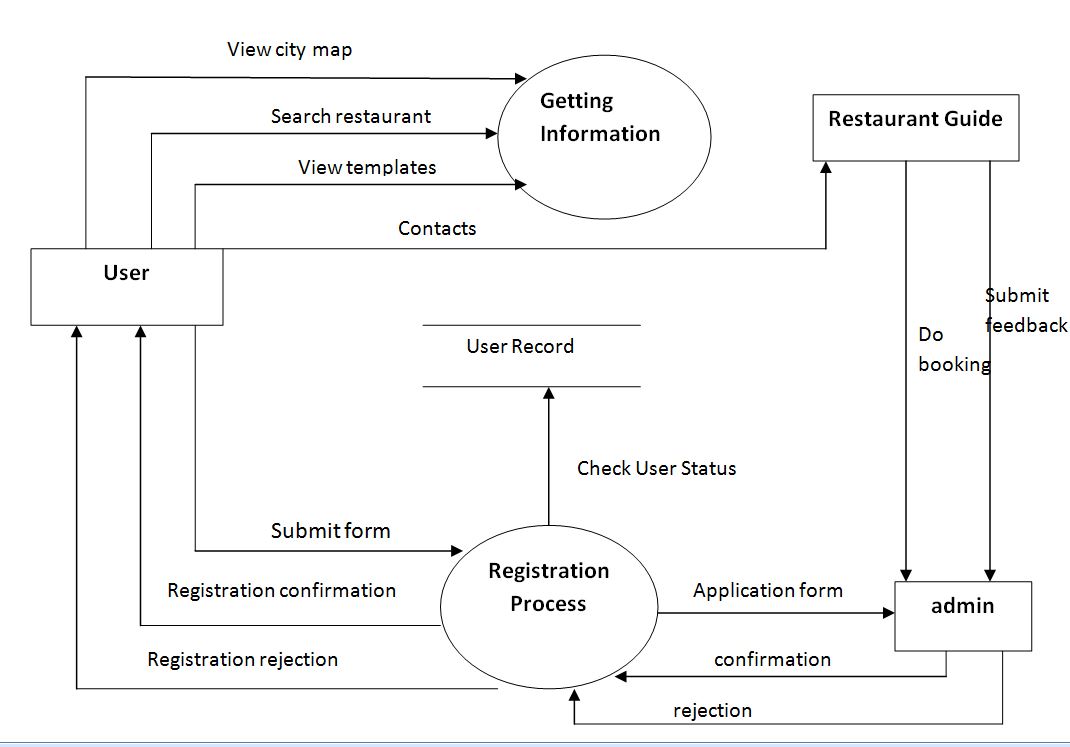
This could also be software-specific like, “New data tables created for this system must be installed on the company’s existing DB server and populated prior to system activation.” Any equipment the customer would need to buy or any software setup that needs to be done so that your system will install and operate correctly should be documented here.

**4 STRUCTURED ORIENTED ANALYSIS:   
Data flow analysis:**

0 –level DTD:

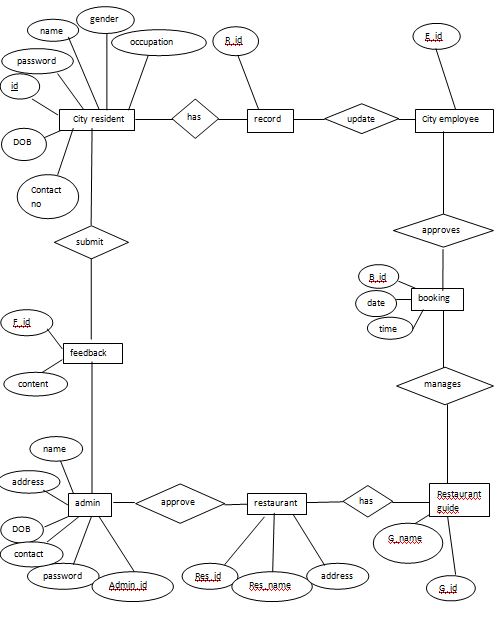


**1-level DTD:**



**5. DATA ORIENTED ANALYSIS:**

**E-R diagram**



**6 OBJECT ORIENTED ANALYSIS:**

## Use case diagrams:



**7. Appendix: User Screens:**

Screen1:Login



Screen 2: Register



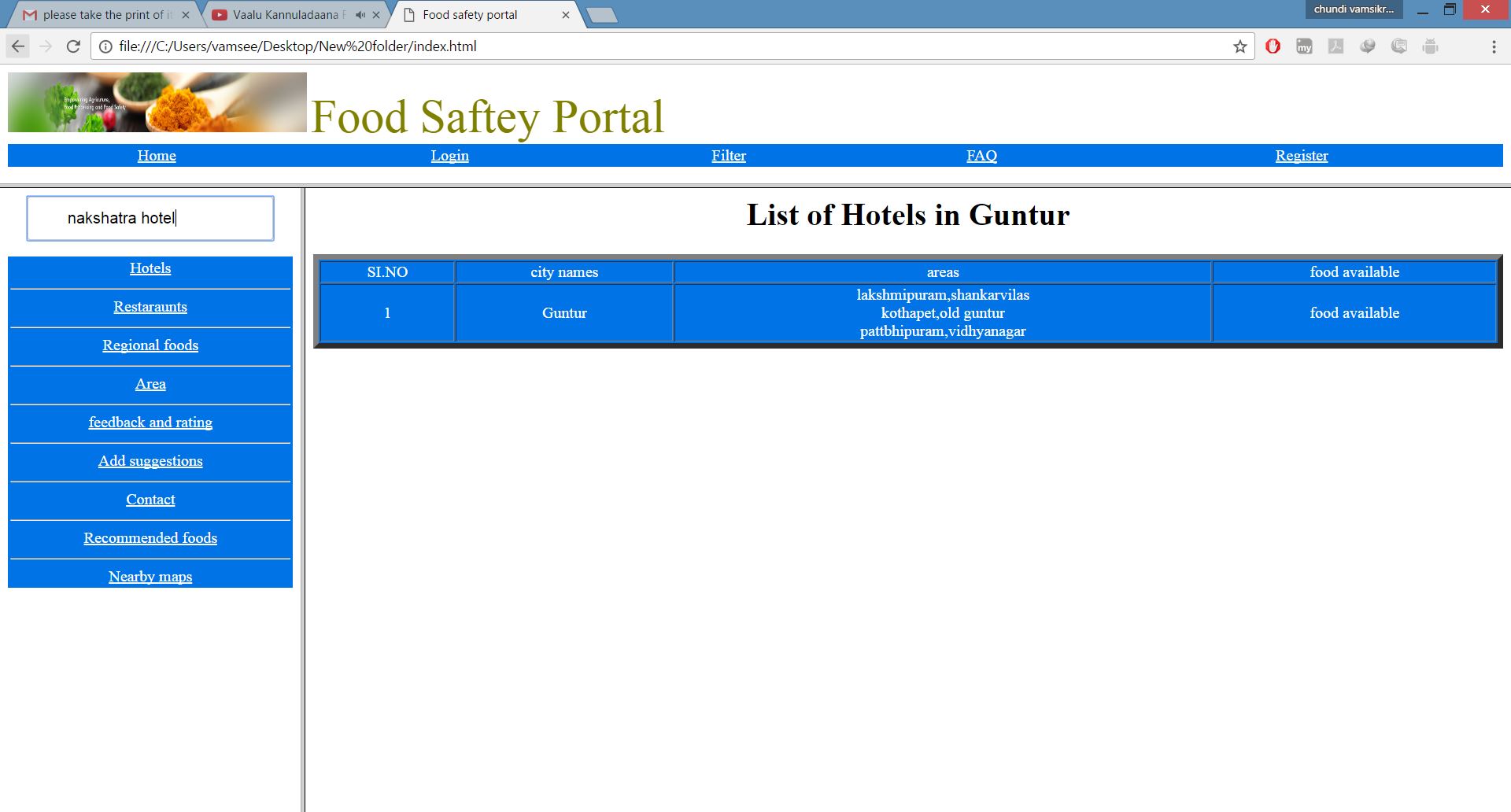
Screen 3: View of database of hotels



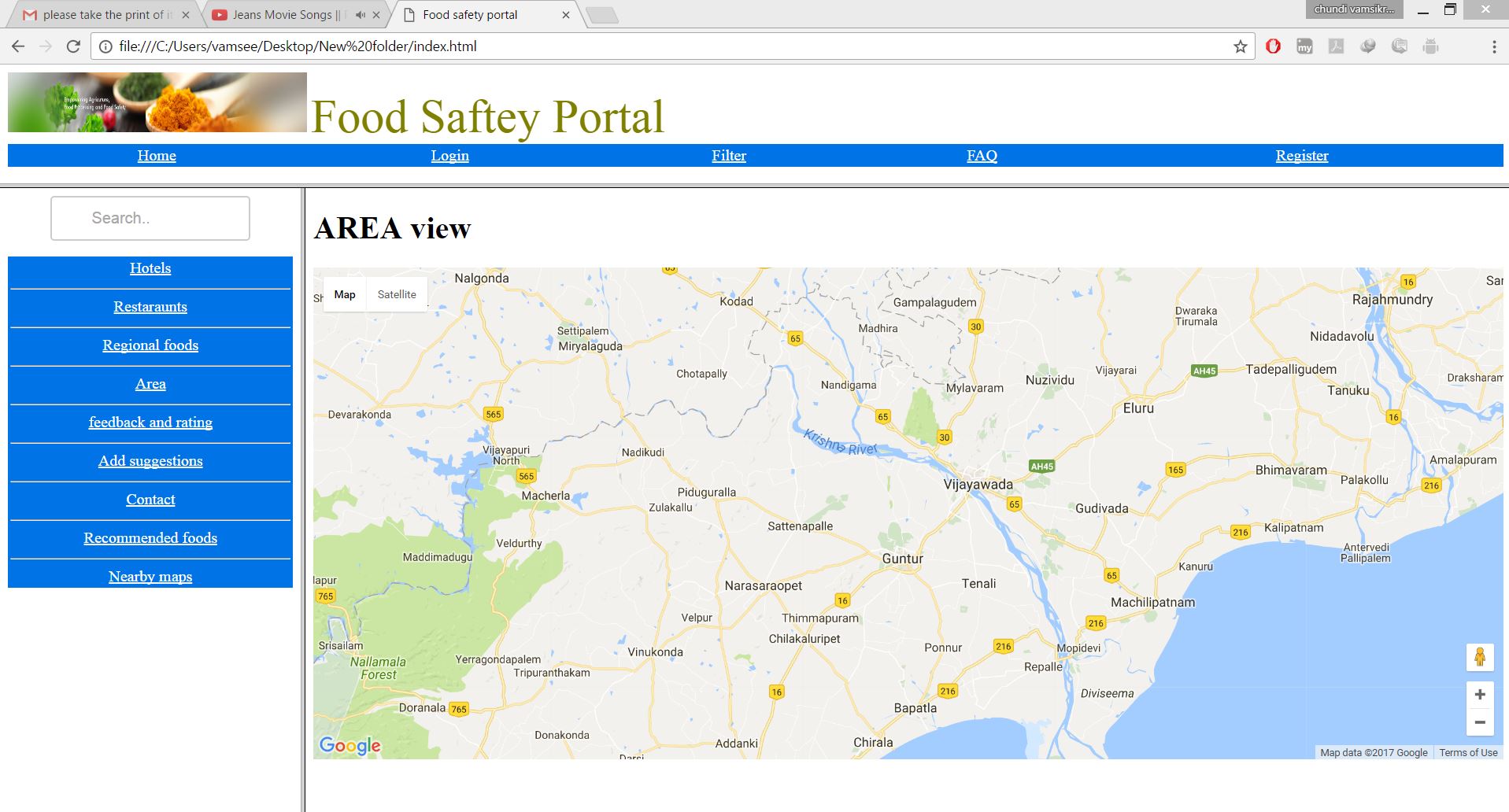
Screen 4: Food center database



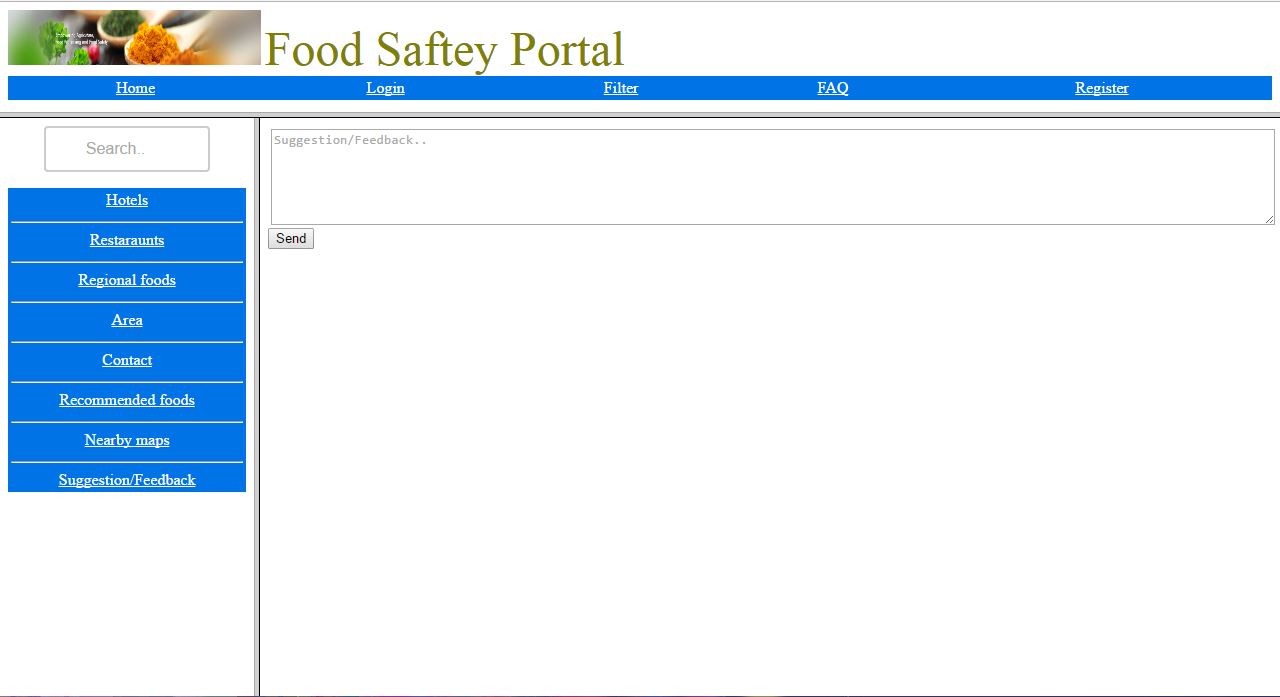
Screen5: Search



Screen6: google maps(location)



Screen 7: Suggestion and feedback



Screen 8: Filter

