**Food Safety Portal**

**1. Introduction**

1.1 Purpose

**Food Safety Portal** is 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 activities and 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 the complains and feedback of users and takes necessary actions.

**1.3 Definitions, Acronyms and Abbreviations**

**1) FSP**

**Food Safety Portal:** It’s a web application that provides online food related services for people.

**2) RAD**

**Rational Application Developer** is a development tool that helps to design web pages and also helps to design the diagrams like ER, Database schema diagrams and to generate DDL.

**3) Administrator**: He has the authority to add/delete users, grant permission to city employees and restaurant guide.

**4) RG**

**Restaurant Guide**: He manages bookings of restaurants.

**5) WASCE**

**WebSphere Application Server Community Edition:** It is an application server that run and supports J2EE and web service applications.

**6) DB2**

**Database\_2**: A database management system that provides a flexible and efficient database platform to maintain records of restaurants, customers, admin.

**7) JSP**

**Java Server Pages:** It is used to create dynamic web content.

**8) J2EE**

**Java 2 Enterprise Edition:** A programming platform which is a part of java platform for developing and running distributed java.

**9) UML**

**Unified Modeling Language** is a standard language for writing software blueprints. The UML may be used to visualize, specify, construct and document.

**10) XML**

**Extensible Markup Language** is a text based format that let developers describe, deliver and exchange structured data between a range of applications to client for display and manipulation.

**11) HTTP**

**Hypertext Transfer Protocol:** It’s a service protocol.

**1.4 Technologies to be used**

* DB2: Relational Database Management System.
* RAD: Rational Application Developer.
* WASCE: Websphere Application Server Community Edition.
* Rational Software Modeler.

**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.
* User reviews and ratings for restaurants*.*

**2. Overall Description**

**2.1 Data Flow Diagram**

**0-Level DFD**

Application form

Validate user

Food Safety Portal

Admin

do registration /login

user

View map

View templates

Confirmation

Search restaurant

In 0 level data flow data diagram user will request for registration or login (if already registered). Non registered users can search restaurant and restaurant guide.

**1-Level DFD**

View city map

**Getting Information**

**Restaurant Guide**

Search restaurant

View templates

Contacts

**User**

Submit feedback

Do booking

User Record

Check User Status

Submit form

**Registration Process**

Application form

Registration confirmation

**admin**

confirmation

Registration rejection

rejection

**2.2 E-R Diagram**

record

City employee

City resident

has

submit

feedback

admin

approves

restaurant

has

manages

Restaurantguide

booking

approves

update

**2.3 Database Design**

**Tables**

**User Table**

|  |  |  |  |
| --- | --- | --- | --- |
| Serial No | Field name | Data type | Constraint |
| 1 | User Id | Varchar | Primary key |
| 2 | User name | varchar |  |
| 3 | Password | Varchar |  |
| 4 | Security Question | Varchar |  |
| 5 | Secqurity Answer | Varchar |  |
| 6 | Email id | Varchar |  |
| 7 | Contact no | Numeric |  |
| 8 | Address | Varchar |  |
| 9 | City | Varchar |  |
| 10 | DOB | Date |  |

**Admin Table**

|  |  |  |  |
| --- | --- | --- | --- |
| Serial No | Field name | Data type | Constraint |
| 1 | Admin Id | Varchar | Primary key |
| 2 | Admin name | Varchar |  |
| 3 | Password | Varchar |  |
| 4 | Security Question | Varchar |  |
| 5 | Secqurity Answer | Varchar |  |
| 6 | Email id | Varchar |  |
| 7 | Contact no | Numeric |  |
| 8 | Address | Varchar |  |
| 9 | City | Varchar |  |
| 10 | DOB | Date |  |

**Restaurant Table**

|  |  |  |  |
| --- | --- | --- | --- |
| Serial No | Field name | Data type | Constraint |
| 1 | Rest Id | Varchar | Primary key |
| 2 | Rest name | Varchar |  |
| 3 | Contact no | numeric |  |
| 4 | Address | Varchar |  |
| 5 | City | Varchar |  |

**Restaurant Guide Table**

|  |  |  |  |
| --- | --- | --- | --- |
| Serial No | Field name | Data type | Constraint |
| 1 | Guide Id | Varchar | Primary key |
| 2 | Guide name | Varchar |  |
| 3 | Password | Varchar |  |
| 4 | Security Question | Varchar |  |
| 5 | Secqurity Answer | Varchar |  |
| 6 | Email id | Varchar |  |
| 7 | Contact no | numeric |  |
| 8 | Address | Varchar |  |
| 9 | City | Varchar |  |
| 10 | DOB | date |  |

**Template Table**

|  |  |  |  |
| --- | --- | --- | --- |
| Serial no | Field name | Data type | Constraint |
| 1 | Name | Varchar | Primary key |
| 2 | Type | Varchar |  |
| 3 | Size | Varchar |  |
| 4 | Location | Varchar |  |

**Booking Table**

|  |  |  |  |
| --- | --- | --- | --- |
| Serial No | Field name | Data type | Constraint |
| 1 | Booking Id | Varchar | Primary key |
| 2 | User id | varchar |  |
| 3 | guide id | Varchar |  |
| 4 | Res id | Varchar |  |
| 5 | guide rates | numeric |  |
| 6 | Date | Date |  |
| 7 | Time | Date |  |

DIFFERENT CLASSES NEED TO BE DEFINED AND EACH CLASS IS CONNECTED TO OTHER CLASS ACCORDING TO THE FUNCTIONALITY.

C **City Resident Class**: It has attributes like user id, user name, password, email-id, contact no, address, city and date of birth. The class city resident is conncted to restaurant class, template class and feedback class for booking a restaurant, to use templates and to give feedback about the restaurant and restaurant guide.

**Restaurant Class**: Attributes of reataurant class are id, name, address, contact no, and password. The functions it performs are booking(), review() and update(). It is connected to city resident and restaurant guide with M:M cardinality.

**Restaurant Guide Class**: Its attributes are id, name, contact no, email-id , address, password and rate. It has functionality of set up a deal between user and restaurant.

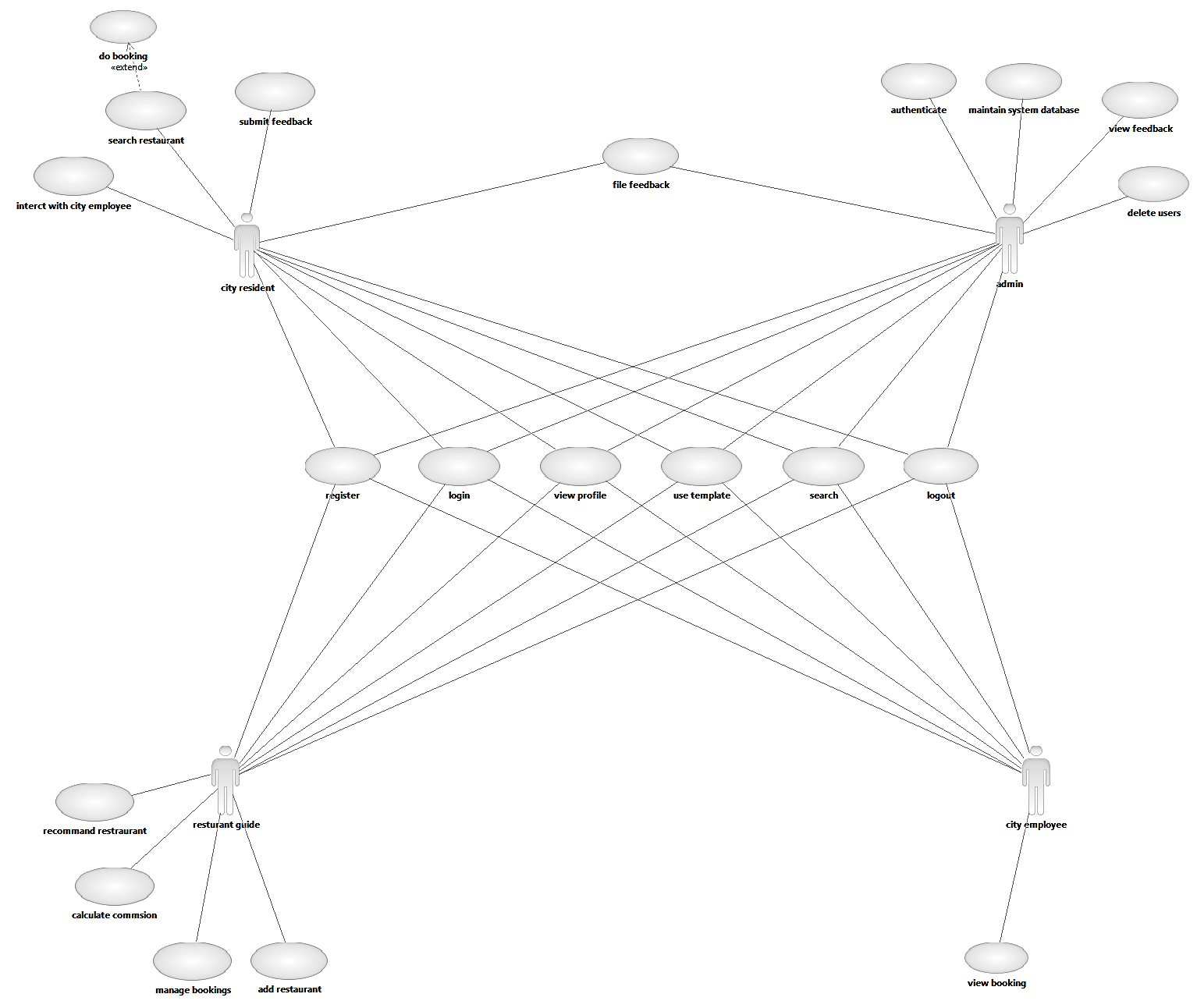
**Admin Class**: Admin is administrator who can look at all the activities which are goinng on. Its attributes are id, name, password, email-id, contact no, address, city. A feedback form is submitted to the admin and according to the feedback he can rate the restaurants and restaurant guide.

**Template Class**: Template class has attribute like type of that template, file name, file size. This class is connected to the users. Users can view the template which are based on food safety.

**Feedback Class**: Feedback class has attribute id, date, subject, content and name of the user who submitted the feedback. It is associated to the user with M:1 cardinality that means one user can submit more then one feedback.

**Visitor Class**: Visitor can visit the website without logging in. They can use the templates and can also see the city map for finind the location of restaurants.

**2.5 Use Case Diagram**



**DESCRIPTION**

**Sign In :**

The user has to Sign In in order to .

**View profile:**

Every registered user has his/her own profile containing personal details.

**Update profile:**

The user has the option to update his/her own profile.

**Select Restaurant or Restaurant Guide :**

The user can select restaurant based on various criteria.

**Do Booking:**

The user can request for booking to a particular restaurant.

**Interact with restaurant guide:**

The user can interact with guide .

**Feedback :**

The user can submit feedback on restaurant or restaurant guide to admin.

**Search :**

The user can search for a particular item in website by entering the keyword.

**3. Specific Requirement**

**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

**3.2 NON-FUNCTIONAL REQUIREMENTS:**

* Because the design patterns of the Online Ordering System are pretty much the standard for a web application, the non-functional requirements of the system are very straightforward. Although written using Google Web Toolkit, the application is cross-compiled to HTML and JavaScript, along with a PHP backend, all of which are supported by any reasonably well maintained web server, although I would recommend Apache2, and particularly the free XAMPP distribution.
* All of the application data is stored in a PostgreSQL database, and therefore a PostgreSQL server must also be installed on the host computer. As with Apache2, this software is freely available and can be installed and run under most operating systems.
* The server hardware can be any computer capable of running both the web and database servers and handling the expected traffic. For a restaurant that is not expecting to see much web traffic, or possibly doing only a limited test run, an average personal computer may be appropriate. Once the site starts generating more hits, though, it will likely be necessary to upgrade to a dedicated host to ensure proper performance. The exact cutoffs will need to be determined through a more thorough stress testing of the system.