1. **Problem Definition:**

Retailer Recommender Using Geo-Spatial Data.

1. **Literature Survey:**

The existing system revolves around the E-Commerce business module while the traditional retailer market business module has suffered a major setback. The main reason for this being that the E-Commerce module enables the shop to reach the customer effortlessly. As a solution, some systems exist that provide retailer coupons and points. But these systems are not robust enough to provide the user with customized and personalized recommendations. Also, most of these systems exist as web-sites and thus have the drawback, that, these systems are not useful to a customer unless the customers visit the web-site themselves. Also, none of these systems gives recommendations taking the users location and history into consideration.

1. **Platform Choice:**

**3.1) Server side-**

Scripting languages (JavaScript, PHP, XML), SQL

**3.2) Client side-**

Java/Objective C

**3.3) Tools-**

- Eclipse

- NetBeans

- MySQL Server

1. **SRS-**

**1. Introduction**

1.1 Purpose.

1.2 Definitions, Acronyms and Abbreviations.

1.3 Technologies to be used.

1.4 Overview.

**2. Overall Description**

2.1 Use-Case Diagram.

2.2 Activity Diagram.

2.3 Sequence Diagram.

2.4 State Diagram.

2.5 Data Flow Diagram.

2.6 Class Diagram.

2.7 Collaboration Diagram.

2.8 Assumptions and Dependencies.

**3. Specific Requirements**

3.1 Hardware requirements.

3.2 Software Requirements.

1. **Introduction –**
   1. **Purpose –**

RRUGD is today’s answer to the E-Commerce market that has almost killed the traditional retail market. Location based social networks provide their users with personalized recommendations. Such a system essentially demands huge amount of data analysis so that the users can be provided with better and more personalized recommendations. The data acquired during these transactions, is extremely useful in behavioral analysis of the users and usage trends. Taking this into consideration, one may use multiple algorithms to carry out this analysis. The algorithm essentially revolves around the analysis of the users behavior and enables prediction of the future possibilities in this pattern. Thus analysis of such data may provide benefits in two ways-

1. It may provide better and more personalized recommendation to the user based on their personal history.
2. It may provide better analytical results to the retail businesses in order to track their sales patterns and further planning.
   1. **Definitions, Acronyms and Abbreviations-**
3. **RRUGD –** Retailer Recommendation Using Geo-Spatial Data.
4. **User-** A normal user who wants to find the nearby retailers.
5. **Retailer-** A retailer who wants to attract users by giving offers and raises business.
6. **Scan Location-** A module which scans the current location and provides with the list of available retailers.
7. **Personalized Service-** A module which means the service if fully personalized which provides retailer recommendations considering user favorites (places and category), personal history and current trends.
   1. **Technologies to be used-**

**3.1) Server side-**

Scripting languages (JavaScript, PHP, XML), SQL

**3.2) Client side-**

Java/Objective C

**3.3) Tools-**

- Eclipse

- NetBeans

- MySQL Server

* 1. **Overview-**

**Existing System:**

The existing system revolves around the E-Commerce business module while the traditional retailer market business module has suffered a major setback. The main reason for this being that the E-Commerce module enables the shop to reach the customer effortlessly. As a solution, some systems exist that provide retailer coupons and points. But these systems are not robust enough to provide the user with customized and personalized recommendations. Also, most of these systems exist as web-sites and thus have the drawback, that, these systems are not useful to a customer unless the customers visit the web-site themselves. Also, none of these systems gives recommendations taking the users location and history into consideration.

**Drawbacks:**

* No personalized service.
* Not considered the user favorites, history and current trends.
* Not location based.
* Have to visit web-site each time for transactions.

**Proposed System:**

* Totally personalized service.
* Location based suggestions.
* Personalized recommendation considering user favorites, history and current trends.
* No need to visit web-site each time.
* More robust, secure and flexible than existing system.

1. **Overall Description-** 
   1. **Use-Case Diagram-**
2. **User-**
3. **Retailer-**
   1. **Activity Diagram-**
4. **Registration Module-**
5. **Recommendation Module-**
   1. **Sequence Diagram-**
   2. **State Diagram-**
   3. **Data Flow Diagram-**
6. **Level 0 DFD-**
7. **Level1 DFD-**
   1. **Class Diagram-**
   2. **Collaboration Diagram-**
   3. **Assumptions and Dependencies-**

* Interface is only in English.
* Login and password is used for the identification of users.
* Only registered users and retailers will be able to use the service.
* The service is hosted on a single server.

1. **Specific Requirements –**

**3.1 Hardware Requirements-**

**3.2 Software Requirements-**

* Windows/Linux
* MySQL Server
* Java
* Eclipse
* Netbeans

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Annex A-**

* **Laboratory assignments on Project Analysis of Algorithmic Design:**

1. **Assignment 1-**

**Algorithmic strategies used in the project-**

‘Retailer Recommender Using Geo-Spatial Data’ is today’s answer to the E-commerce market that has almost killed the traditional retail market. Location based social networks provide their users with personalized recommendations. Such a system deals with a huge amount of data including the users’ favorites, their search history and so on. Hence it requires the proper mechanism to manage such a huge data. Thus in our proposed system we are making use of an ‘Apriori Algorithm’.

Apriori is a classic algorithm for frequent item set mining and association rule learning over transactional databases. Apriori uses a Dynamic Programming strategy since it deals with huge amount of data, frequent item set and association rules. RRUGD is completely a real time service, the data in the real time gets changed frequently, thus to deal with such frequently changing dynamic data, the dynamic programming strategy is best suited.

It is based on the Breadth-First-Search strategy. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database. The frequent item-sets determined by Apriori can be used to determine association rules which highlight general trends in the database.

1. **Assignment 2-**

**Why given option is chosen?-**

Considering and analyzing the today’s existing service in the E-Commerce domain, we realized that E-Commerce has changed the customer’s view of shopping and making transactions. Most of the people now-a-days have preferred online buying and selling their goods or services. Although it has many advantages over traditional market, but as there are always two sides of a coin, it has some drawbacks too. The major drawback of such a system is customer has to visit web-site each time for doing transaction (buying or selling). Another major drawback is current system does not provide customer the personalized service considering their needs, favorites or history. Thus we decide to select this option that will provide customers the personalized service and next location recommendations considering their favorites, needs and history.

1. **Assignment 3-**

**Modules of the projects, the students are developing-**

|  |  |
| --- | --- |
| Module | Student developing |
| 1. Registration | Mr. Nirmik Kale |
| 1. Recommendation | Mr. Kishan Pawar |
| 1. Offer-Checkin | Mr. Abhishek Kulkarni and  Mr. Jigar Panchal |

1. **Registration Module-**

This module is responsible for registering the new users and retailers to the RRUGD service.

1. **Recommendation Module-**

This module is responsible for suggesting the nearby registered retailers when user scans the location.

1. **Offer-Checkin Module-**

This module is responsible for availing a specific offer which user checkin and generates a an unique token for availing and actual transaction purpose.

1. **Assignment 4-**

**Domain specific analysis-**

Now-a-days an E-commerce has completely killed the market of traditional retailers, thus the solution to this problem is RRUGD i.e., ‘Retailer Recommender Using Geo-Spatial Data’. Location based social networks provide their users with personalized recommendations. Such a system essentially demands the huge data in the form of their location etc. The major factor of data is location which is required for suggesting the next retailer location based on the users’ history and his/her favorites. Hence as per the analysis of this data we concluded that the next retailer location can be suggested by using all this data. Thus while suggesting next retailer location the frequency of an user visits to a particular location will also be considered for providing better suggestions. Henceforth we are using an ‘Apriori Algorithm’ for the same, which works on the huge amount of data-sets and provides the frequent data-sets according to the conditions. Hence it will be useful for retailers suggestions based on users’ location since it deals with frequency.

**Annex B-**

* **Testing, Reliability parameters and test plan.**

Our main objective of testing is to determine causes of failure for our project. By testing we are able to detect the bugs, defects and errors that cause service to fail. As we know that every service or application cannot be 100% bug free, hence we will try to find as many bugs, defects and errors as possible to make our service bug free and quality service.

**Methodologies –**

1. **Unit Testing –**

Unit testing is performed on the small modules of a project. Since testing is done on modules it is also called as module testing. Thus we will be performing Unit Testing on the following modules of RRUGD-

|  |  |  |
| --- | --- | --- |
| Module | To test | Criteria |
| 1. Registration | Input fields | Should fill mandatory field. |
| * 1. User registration. | Input fields | Should fill Mandatory fields. |
| * 1. Retailer registration. | Input fields | Should fill Mandatory fields. |
| 1. Token generation. | Generated token | Should be according to pre-defined format. |
| 1. User location scan | User exact location | Should scan exact location. |
| 1. Retailer location scan | Retailer exact location | Should scan exact location. |
| 1. Offers | Available offers | Should be returned exact offers to the same respective user of respective retailer. |

1. **System Testing-**

System testing is performed for checking the overall performance and functionality of a service or a system. The system is tested as a whole in the system testing. Hence in case of RRUGD, we will check the performance of a service as a whole by considering following factors-

1. How the service will response at the peak time i.e., multiple requests at a time?
2. Does service suggest appropriate locations after check-in?
3. Does service exactly relates the specific users’ history and favorites while location prediction?
4. **Validation Testing-**

Validation testing is performed for the validation purpose of the functionalities of a service or a system. Thus in case of RRUGD validation is in the form of modules that are specified in the context of Unit Testing.

1. **Integration Testing-**

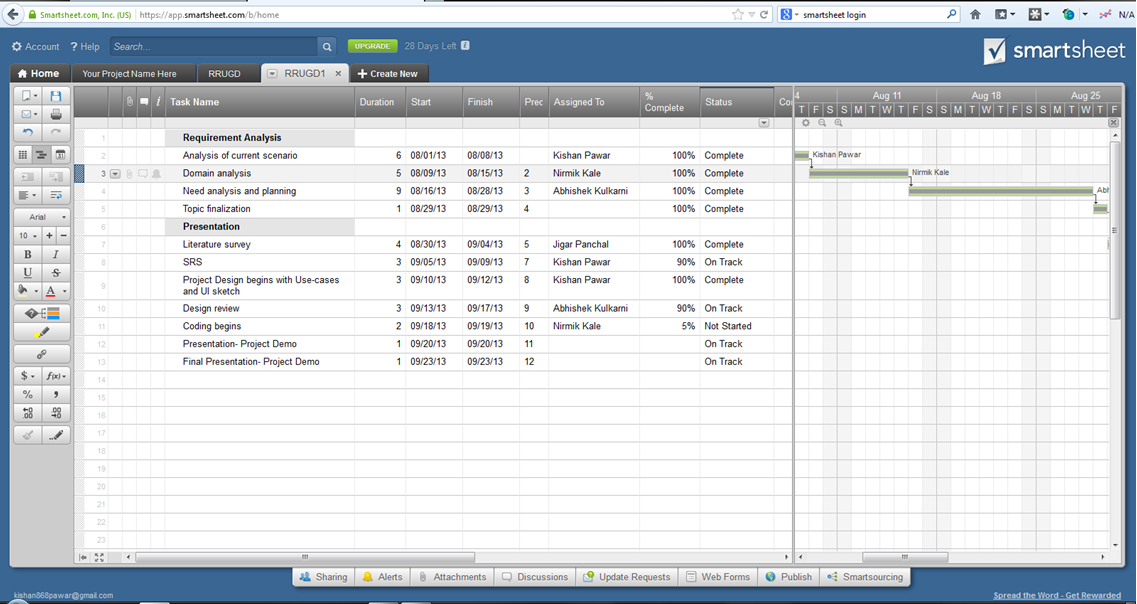
Integration Testing is performed for testing the service or a system for its functionality when the modules of a service are integrated in a single large package. The integration testing focuses on how the service will work when all the modules are integrated in a complete service or a system as a whole. Thus in case of RRUGD we will test the complete integrated service is suggesting the exact location according to specific users’ history and his/her favorites.

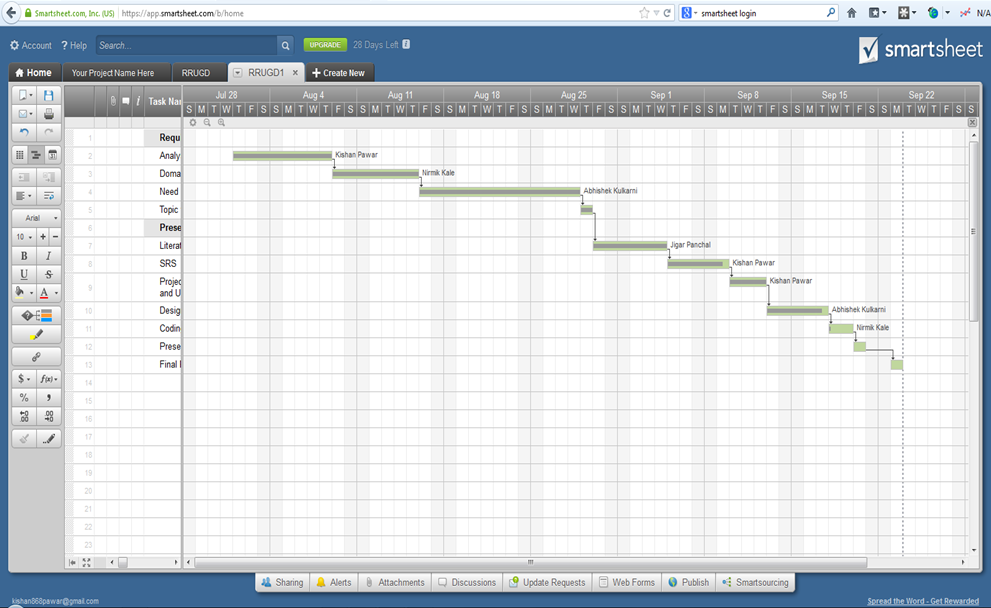
**Testing Strategy-**

We will be using the Dynamic WhiteBox Testing strategy since we are not using any automated tool which provides some degree of automated inbuilt testing. Since we are designing the service from the scratch with scripting languages we will have to be concerned about all the functionalities and features provided by the service while developing the same. Hence dynamic WhiteBox testing strategy allows us to be concerned about how the internal functions are getting executed according to the conditions of the real world time.

**Annex C-**

* **Planner and Progress Report.**

****

****