CSc 322 Software Engineering Lab

Fall 2017

A product ordering system for a warehouse

This project could be done entirely as a web application or local system. You are neither required nor encouraged to develop a web or mobile system. You are free to make your own choice for cases/situations/features that are not described in this requirement list.

Users of this system with different authorities:

1. *Client* – owners who buy products from salespeople. They can rate salespeople they purchased products. And they are also rated by salespeople who sold products to them. Clients and salespeople can file complaints against each in the system. Any clients complained twice are put in the black list and can never log in to the system. A salesperson complained twice will receive only 90% commissions. The salespeople who were complained 9 times in total are suspended until the manager decides otherwise. They can also press "like" or "dislike" to each other. The clients receiving 3 likes/dislikes will receive 5% discount and 5% price increase, respectively; the salespeople netting every 3 likes/dislikes will receive 5% more/less commissions. Clients and salespeople who pushes "like/dislike" for more than 9 times in a row will be punished with money amount more than s/he received. (Which means the history of every user's "like/dislike" should be remembered in the system.
2. *Salesperson* - the person who visits the supermarkets and delis around the city and places orders that go to the warehouse. The salesperson has a small percentage of discount to make the sale and receives a percentage of commissions as defined by the supervisor
3. *Manager* - the person who controls payments and the sales history of the salespeople that are under its responsibility. The supervisor also makes a larger promotion available or authorizes changes in the prices of the items. The supervisor can change the percentage of the commissions of, and fire and hire the salespeople in the same group. Managers can black list a client if something is wrong with the payments.
4. *Director* - the person determines if an item is available to be seen in the app and the amount of cases available. The director can promote salespeople to managers and demote managers to salespeople. The director decides to place more orders of products or stop order any more products based on the sales statistics.

System requirements:

1. Different users should have different personalized page when s/he logs in. For instance, a new client will only see the best 3 selling products all through the warehouse; if s/he just purchased certain products then s/he will see top 3 best-selling products of the that type. For managers the best and worst salespeople's name of their managed groups are shown. The director will see the best and worst performing groups of the entire warehouse.
2. Clients and salespeople can only see their own transactions and statistics; managers can see the salespeople and clients of their groups. The director can see everything.
3. The warehouse inventory is initialized/added by the director, and the system updates it automatically after every transactions. It cannot be negative.
4. All users can search for the clients/salespeople with the worst/best ratings.
5. One creative features (10%) of your own choosing to render this system smarter and/or more appealing to its users.

An E-book sharing system specification

In this system, users can share and review E-books in a new way.

Three types of users:

1. Super-users (SU): all RU's allowed operations, new-user approval decisions, book approvals/updates and complaints processing, set up the points for different reading durations.
2. Registered users (RU): all VS's allowed operations, contribute books, read and rate/review books/reviews, send complaints to SU on book contents.
3. Visitors (VS): browse available E-book catalog and the reviews/ratings, apply to be a new RU.

System features:

1. Provide a catalog of available E-books, each E-book should have a cover-page and summary to be displayed in the catalog, and the number of points needed to read the book for a certain time, e.g., 10 pts for 10 min, 20 for 30 min and 30 for 1 hour (in your system the time units should be seconds instead of minutes to make it easy to test).
2. For any RU to read a book, s/he should have points in the system >= the pts asked by the book for a certain amount of time. Once the RU start to read the book, a timer is set according to the time the RU purchased, the book can be closed by the RU before the purchased time, or the system will close it when the time comes. The duration of the reading time for one book by one RU is accumulative.
3. The RU can review and rate a book only if s/he read it before, each review and rating is weighted by the amount of time the reviewer reads the book—the rating of an RU who reads the book with 2 hours should have a larger confidence that that by another RU who only read it 10 min.
4. The RU can contribute a book to the system and asks for a certain number A of points to be deposited to his/her account (this is the only way an RU can get points), the SU decides the number B of points to be rewarded to the contributing user. If B < A, the RU will be notified for approval, if not, the contribution will be denied.
5. Your system should have a function to search for bad words in the books. Any RU can have their own choice of bad words, the RU can complain the book based on his/her search. An RU can also complain a book after s/he read the book. A book receiving 3 complaints will be removed automatically, the point B for this book when the RU contributed this book will be deducted from his/her account with additional -100 points as penalty. If the complaint is very serious, e.g., copyright violations, the SU can choose to punish the contributing RU even with only 1 complaint. An RU whose books are

removed twice or who contributed one copyrighted book will be ejected from the system and put in the blacklist who can never register again.

6. Every time an RU log into the system, 5 books that are similar to the ones s/he read before will be recommended. If an RU never read any book, then the top five books that were most read in the system will be recommended.
7. An RU can invite another RU to read a book, once accepted, they both can read the book for their purchased time, and they split the asking points.
8. Each RU can check his/her reading history/stats; any one, including a VS, can browse the stats for each book.
9. A book that no one read for a certain time will be removed and 5 points are deducted from the contributing RU.
10. A new creative feature worth 10% of the total project credit based on each team's own choice.

General:

1. Should have one user-friendly GUI holding all features together (don't pop up many windows for different features)
2. The system need not be Internet based, a stand-alone app is fine.