COMP 2920: Software Architecture & Design

Class Activity #4

Objective:

To walk through the process of defining software architecture of a system using UML diagrams

Description:

Let us define a CD sales system that displays the record of sales of a music band. The overall sales are recorded in the external Billboard Reporting Service. Our system co-ordinates with the external system and can show the details of sales of specific band to the band manager. The records manager of our company can view the overall billboard report as well as the details of specific band or other music.

Please note that our system delivers Billboard reports from an external Billboard Reporting Service.

Optional: If you wish, you may add additional services to the system.

To do:

1. Recommend the software architecture pattern used for this system.
2. Design the use case diagram for the system. Since its all about viewing sales reports, login into the system is must.
3. Identify the classes and sketch the class diagram using multiplicities. Also define the common attributes and the functions required to define the system
4. Design a sequence diagram that allows the records manager to generate the report for a particular CD for the specific week.
5. Convert this sequence diagram to the communication diagram.
6. Is there any state chart diagram associated with this system? Draw if any.
7. Design the activity diagram for generating the report of sales for specific CD by the records manager
8. List a few functional and non-functional requirements for this case study
9. List the specific domain requirements for this case study, if any.
10. What software development model would you like to use to develop this system and why?

Instructions:

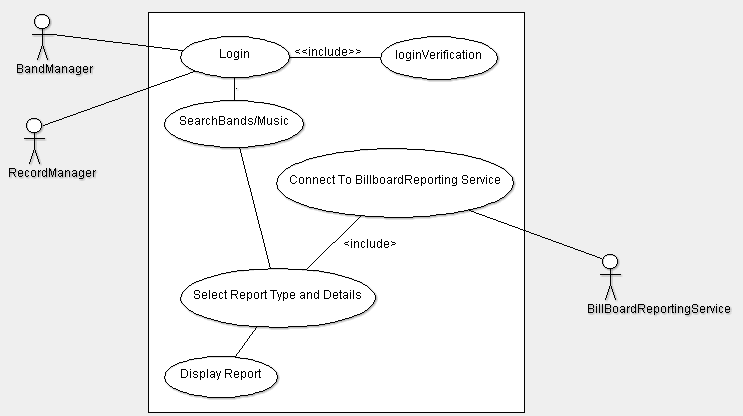
Make a group of two students and start documenting everything you feel important for this case study keeping in mind the 10 questions given above. You are free to change the sequence of questions to answer.

Make sure you make a complete documentation of this case study as I will be needing that to mark this activity.

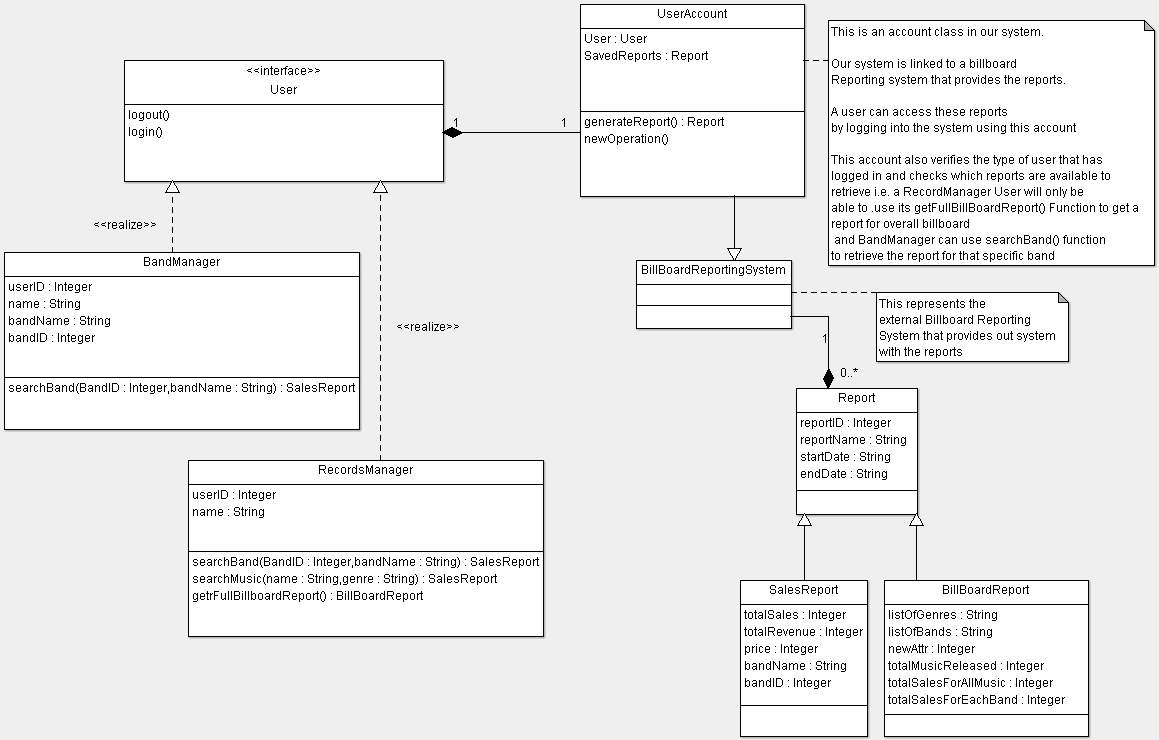
**Solution:**

For this kind of a system, we would recommend using a Client/Server architecture where our system would be a client for the server that is the Billboard Reporting service. Also, more layers could be added to this system such as a database layer to add functionality to save generated reports. This would then be used to retrieve the reports instead of always having to call the Billboard service to always freshly create a new report.

A Use case to retrieve a report from the bill board service:

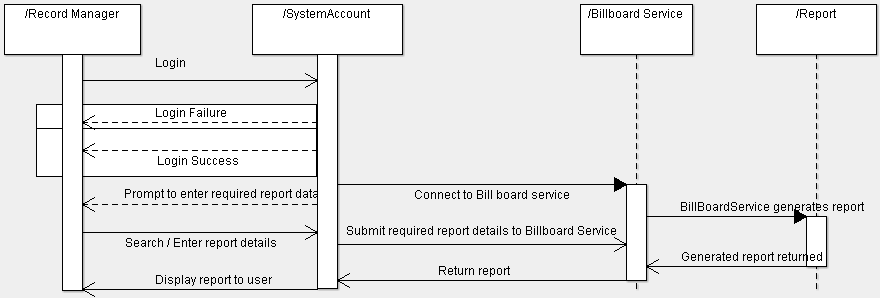


**Class diagram:**



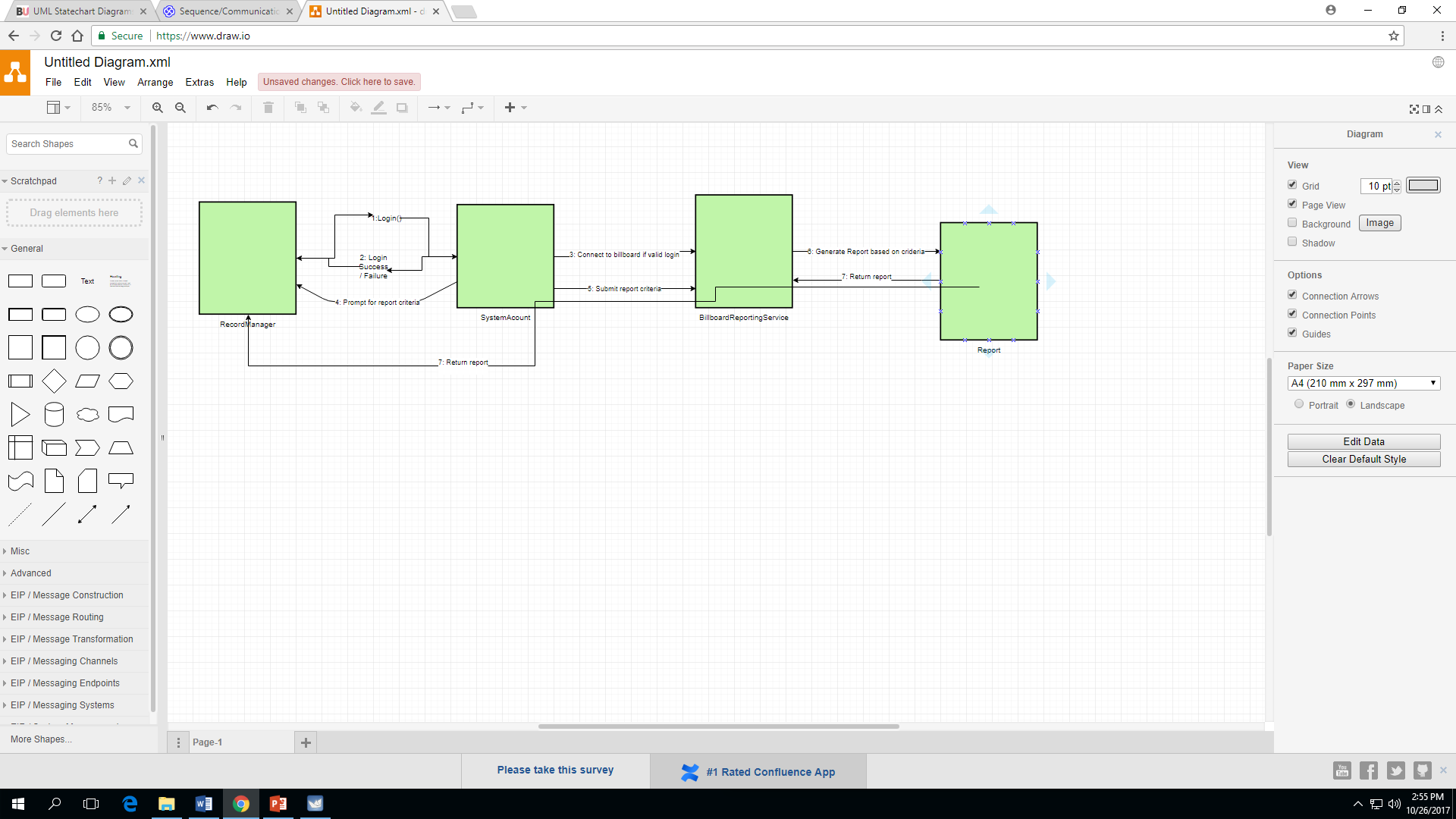
Each user is allocated a unique account in our system. This account is used when connecting to the Billboard system to verify the type of report that a user is allowed to access for example, a RecordManager account will be able to access both Sales and Billboard reports where as a BandManager will only be able to access a Sales report for that specific band.

**Sequence Diagram:**

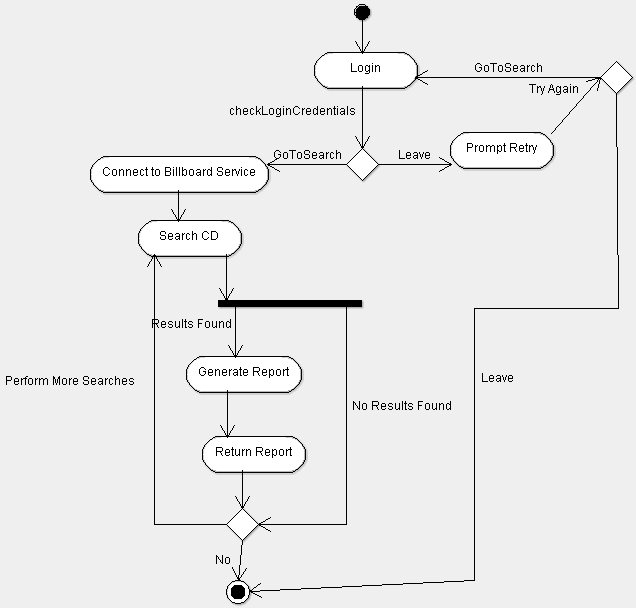


The record manager logs in to the system. If valid credentials are provided, the System then connects to the Billboard Service. The user is then prompted to enter required report details that are then forwarded to the BillBoard service. The billboard service then generates the required report based on the criteria and returns it to the System which then displays it to the user.

**Communication diagram for above scenario:**



**Activity Diagram:**



After a RecordManager logs in to the system, the system checks the validity of the credentials. If they are invalid, the user is prompted to re-enter them and user can also decide to quit at this point instead of re-entering. If the credentials are valid, the System then connects to the BillBoard reporting service and prompts user to enter the data and criteria to send to the Billboard service which will be used to generate and return the report. The system then collects the generated report from the Billboard reporting service and displays it to the user.

User can then decide to quit or perform more searches to generate more reports.

**State Chart Diagram:**   
For User object: there are two major states, Logged In and Logged Out

For the System Account: Connected or Disconnected

**Functional and Non-Functional Requirements:**

|  |  |
| --- | --- |
| Functional Requirements | Non-Functional Requirements |
| Require user login with valid credentials | Allow a max of 5 incorrect login attempts before locking the account |
| Display UI for user to select reports | UI should be easy to read, follow and use |
| Save generated reports and make available to access without getting new one from Billboard reporting system | Report retrieval from Billboard service should take less than 10 seconds |
| Display reports | System should be able to handle multiple users accessing at the same time |
| Multiple types of accounts for example RecordManager account and BandManager account | System should be able to easily change or connect to different Billboard services |
| Restricted access to each account to view specific types of reports |  |
|  |  |

**Domain Requirements:**

* Genre Searches: Users should be able to filter searches according to the Genre of the music/ band
* Band Searches: User should be able to search according to a name or id of a specific band
* Multiple users must be able to use the software simultaneously
* Reports generated must be matching the selected criteria of the user
* System must be able to connect to multiple Billboard Reporting services

For this system, we would recommend the Agile or RAD models. This is because the system is mainly just displaying a report that it receives from the Billboard Reporting System. There are mainly two types of users, BandManagers and RecordManagers and each with main functions of commanding the system to retrieve reports from the Billboard reporting System.

This system is not very large and doesn’t require a lot of detailed documentation thus Agile or RAD would be suitable for this system.