

## PROJECT PART 3

**Team Name and Number:** Movie Recommendation System (#32)

**List of Team Members:** Chaitra Ramachandra, Vipra Gupta

### PROJECT SUMMARY

**Vision:** To build a robust movie recommendation system that can recommend movies to its users by analysing their movie watch patterns and movie preferences.

**Project Description:** Movie Recommendation System is a web application that will recommend movies to its users by analysing their genre preferences and watch patterns. The application provides an interface through which the users can login into the system, update their basic profile, the application will ask them certain questions and build the user's profile. The application will also perform analysis on the user's movie watching patterns and the ratings they gave to them to recommend movies according to the his/her taste.

### CLASS DIAGRAMS

**Previous Class Diagram** can be found at the end of this document.

**Previous Class Diagram - Fixed** can be found at the end of this document.

**Completed Current Class Diagram** can be found at the end of this document.

### SUMMARY

- Redesigned the class diagram to include the suggested changes.
- Added Factory Method Design Pattern to the new class diagram. Other than this, we will be using the Java provided to iterate over the lists.
- Implemented the following classes in Java
  - User
  - Admin
  - Customer
  - Movie
  - Recommend
  - Payment
- Created the Completed Class diagram for the implemented set of classes
- Deciding on database and database tables
- Started work on coding the frontend (UI) for home page

### BREAKDOWN (of Part 3 tasks):

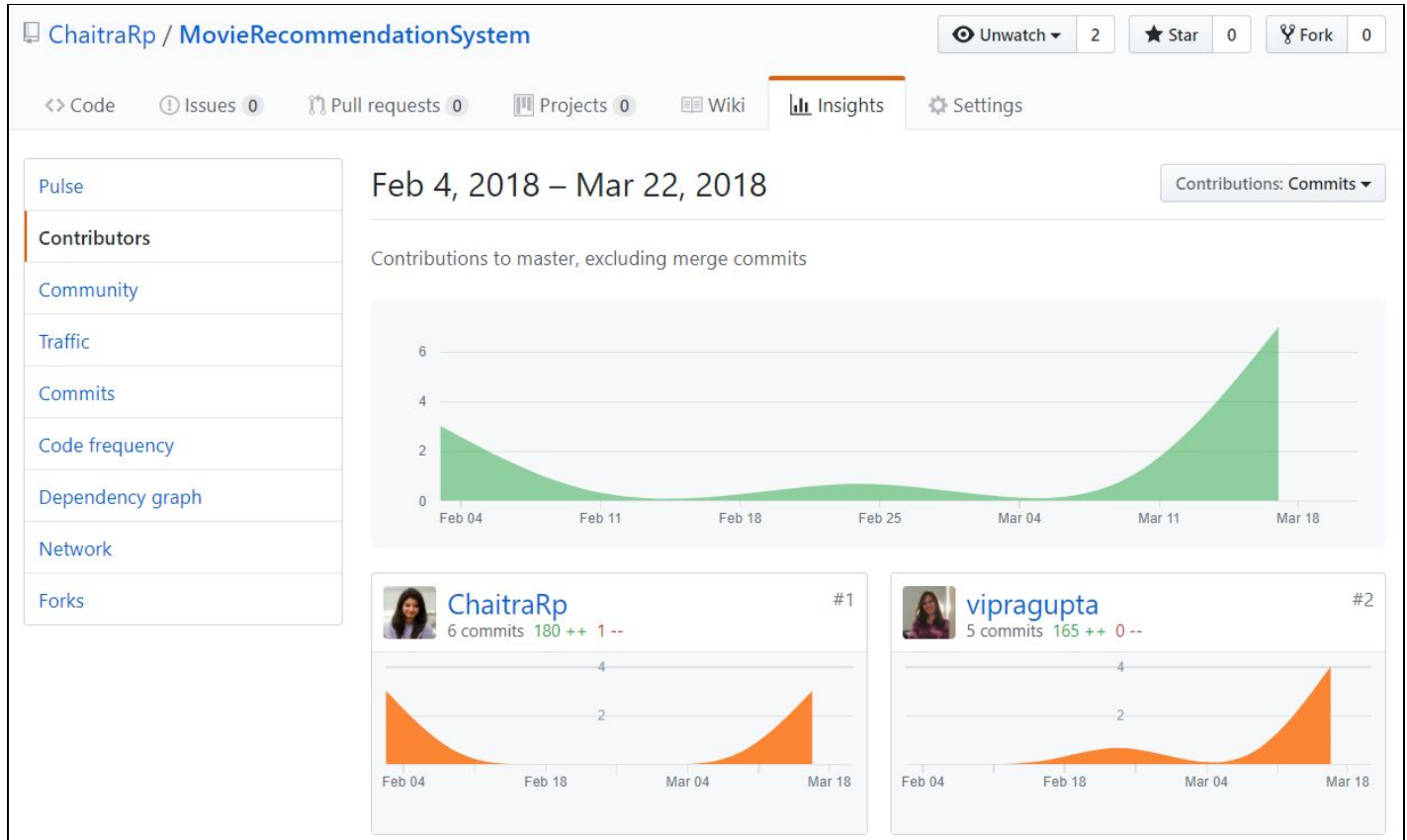
**Vipra Gupta**

1. Fixing previous Class Diagram
2. Current Class Diagram
3. Adding Design Patterns

## Chaitra Ramachandra

1. Database and Database Tables
2. Implementation of classes
3. Research on UI options

## GITHUB GRAPH



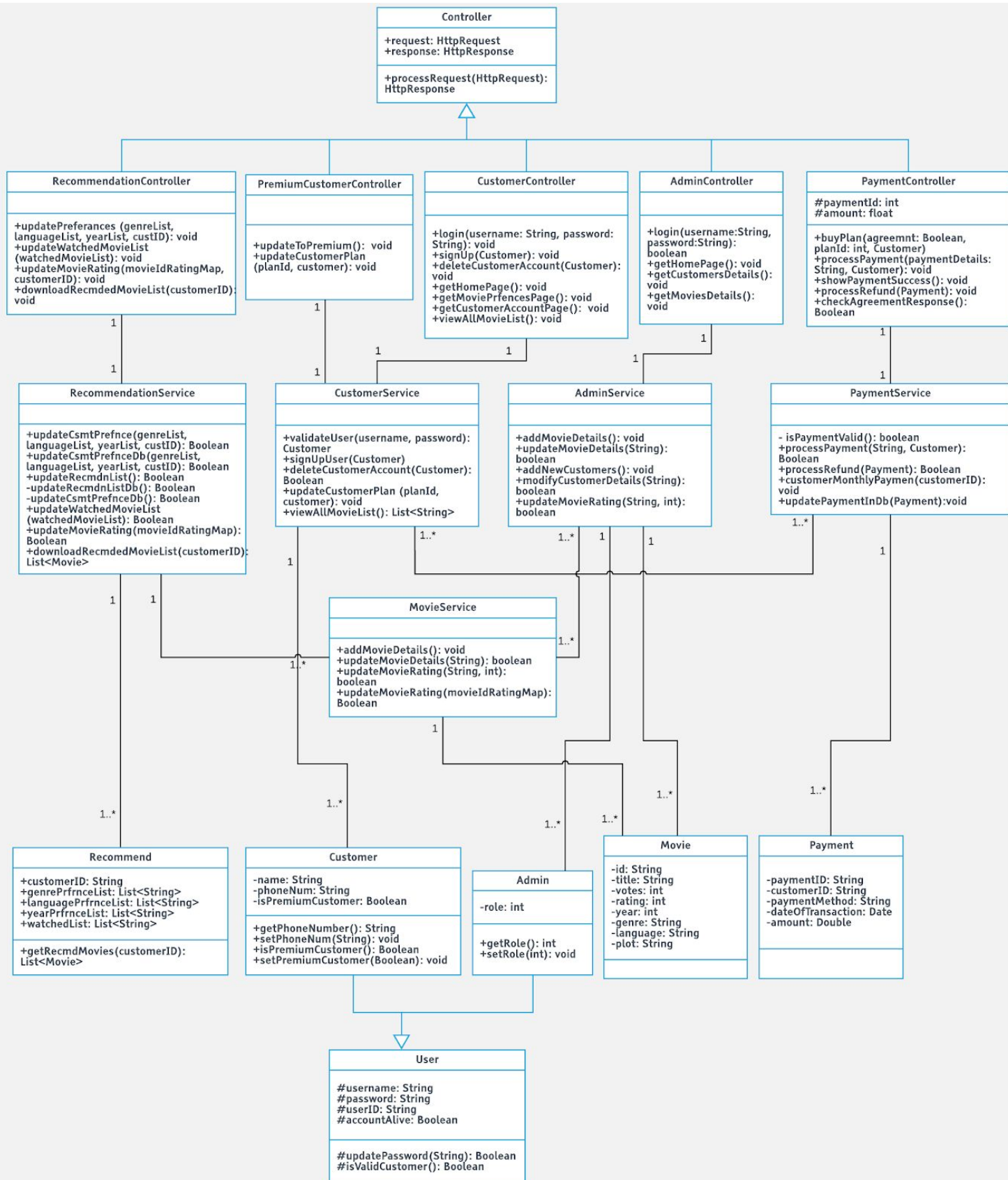
## ESTIMATE REMAINING EFFORT

- 1) Implementation of Login and Session Management - 10 hours
- 2) Implementation of classes and design patterns - 18 to 20 hours
- 3) Creating Unit tests - 3 hours
- 4) Refactoring - 2 hours
- 5) Reflect dynamic changes as the system progresses - 2 hours
- 6) Database implementation and connections - 4 hours
- 7) User Interface - 10 hours

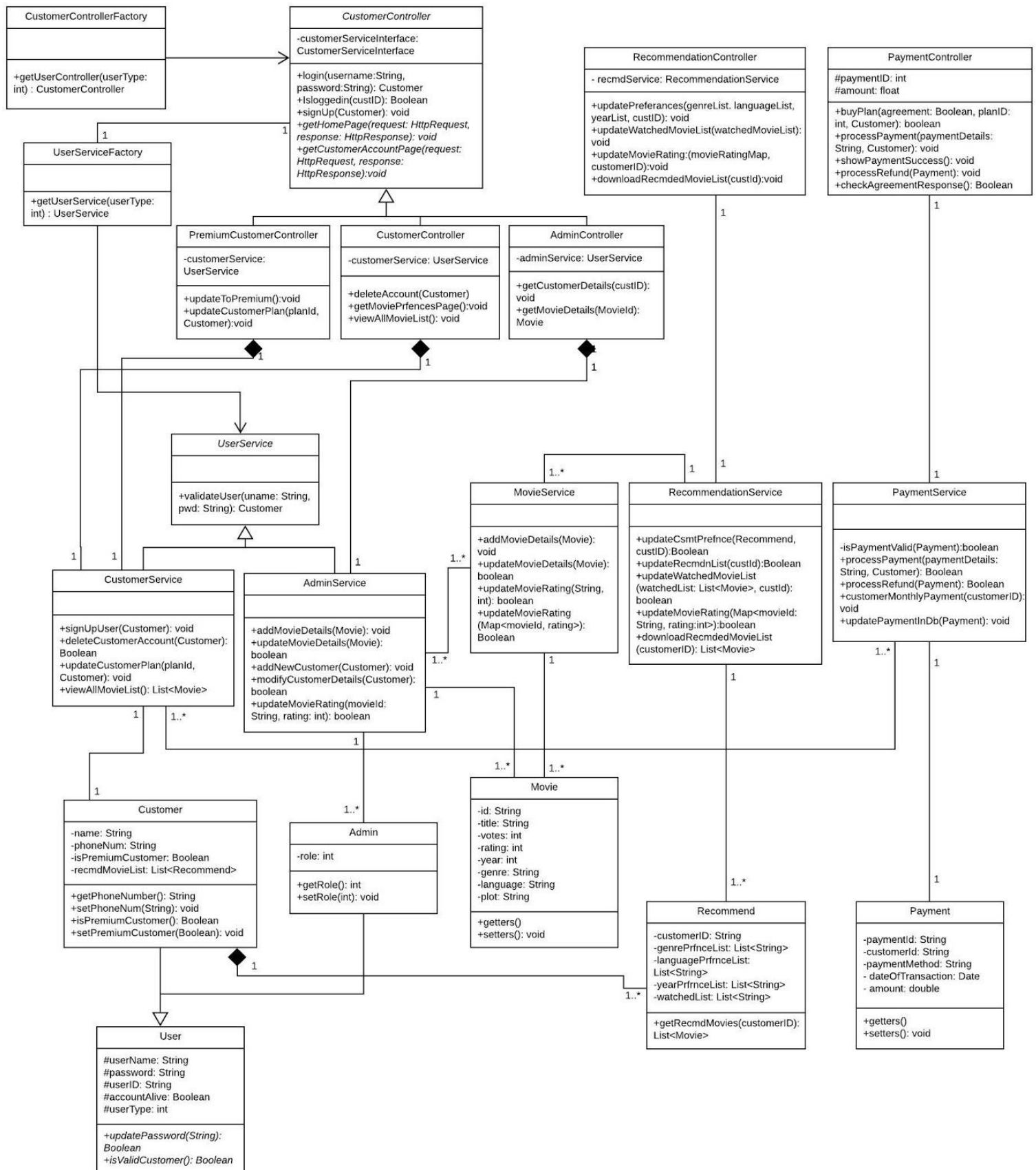
## NEXT ITERATION

- 1) Implement all class files
- 2) Implement Design Patterns
- 3) Implementation of Login
- 4) Writing a few unit tests
- 5) Connecting to Database

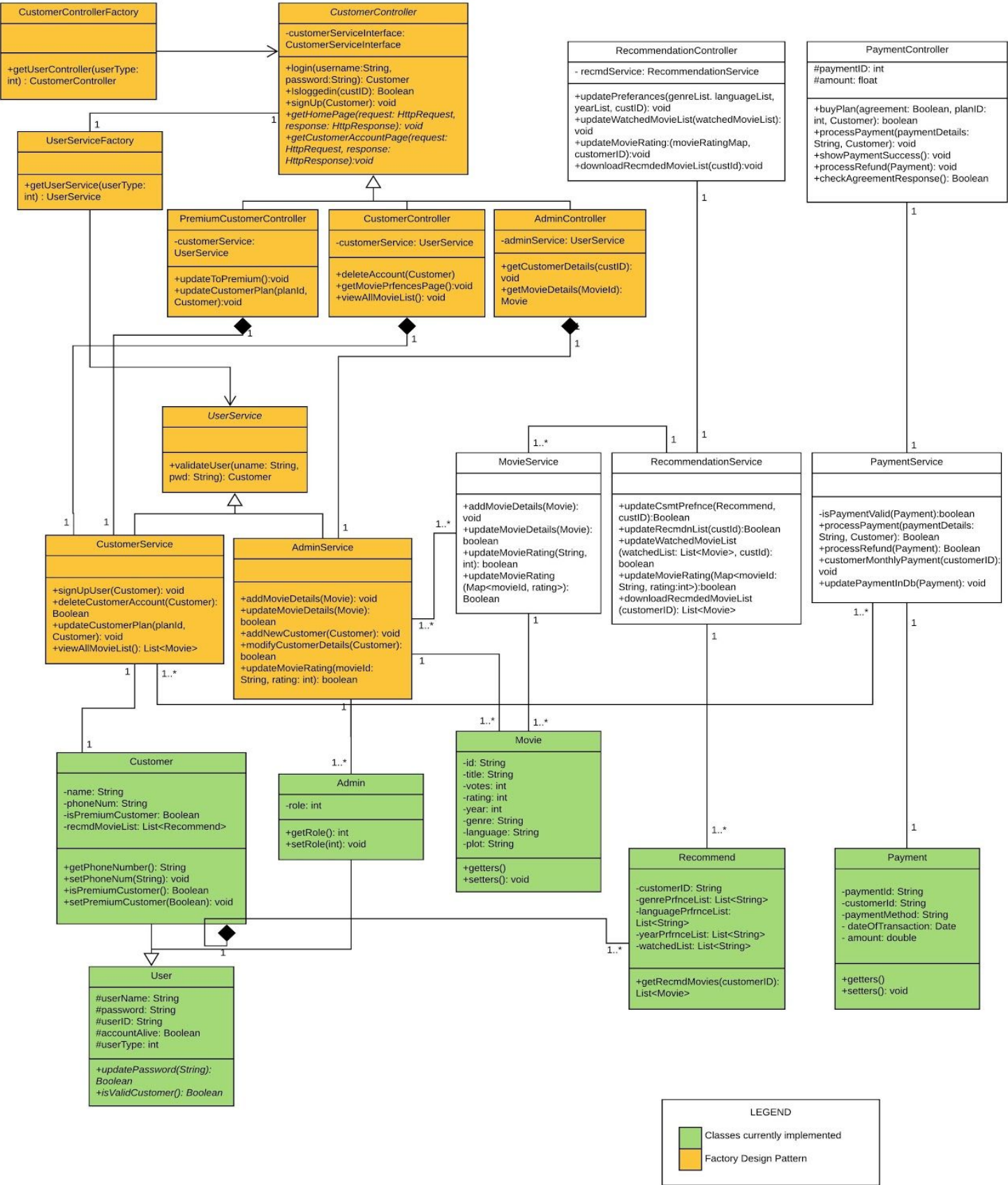
### Old Class Diagram: (Can also be found in git as separate file)



## New Class Diagram: (Can also be found in git as separate file)



Current Completed Class Diagram: (Can also be found in git as separate file)



Factory Design Pattern - Class Diagram: (Can also be found in git as separate file)

