|  |  |  |  |
| --- | --- | --- | --- |
| Functional and non-functional requirement | Requirement | Justification | Moscow |
| F(R1) | User Registration to system | User must registration to system. | Must have |
| F(R2) | User Login to system | User must Login to System for booking movies. | Must have |
| F(R3) | Add Movies | Admin can add movies | Must have |
| F(R4) | View Movies | User can view details of movies | Must have |
| F(R5) | View show time | User can view details of show time of movie | Must have |
| F(R6) | View show date | User can view details show date of movie | Must have |
| F(R7) | Update User profile | User can update their profile | should have |
| F(R8) | Update Movies | Admin can update movies | Must have |
| F(R9) | Ticket generate after booking Movies | Ticket must generate after user book particular movie | Must have |
| F(R10) | Seat selection | User can select seat | Must have |
| F(R11) | Show name | System must view show name | Must have |
| F(R12) | Show type | System must view show type and show time | Must have |
| F(R13) | Delete Movie | Admin can delete movie | Must have |
| F(R14) | Admin Login | Admin can login to system | Must have |
| F(R15) | Update Admin Profile | Admin can update his/her profile | Should have |
| NF(R16) | Security of system | Make sure that System must secure and protected from unauthorized access. | Must have |
| NF(R17) | Portability | System should be run in every platform. | Should have |
| NF(R18) | usability | System should navigation easily and easy to use. | Should have |
| NF(R19) | Scalability | System should store more and more information easily | Could have |
| NF(R20) | Maintainability | Maintenance should be done in regular basic | Must have |
| F(R21) | Check seat availability | Admin and user can check available seat | Must have |
| NF(R22) | Privacy | System should keep user information private | Must have |
| NF(R23) | Reliable | Precise and accurate information should given to system | Should have |

**Use-Case Diagram**

A Use-case is a software and system engineering term that describe how user can interact with system to perform a particular operation. A use case acts as a software modelling technique that implies function to be implemented. (Techopedia, 2018)

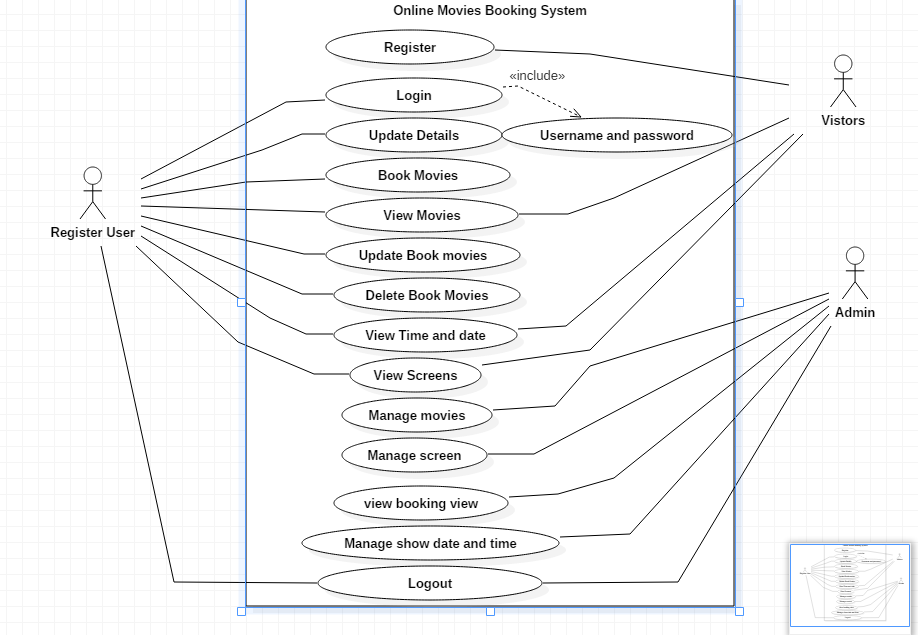


Figure 1 : Use case diagram for online movies booking system

**Class Diagram**

A class diagram is a description of relationship between different classes. It is a static diagram which represent static view of system. It describes the attributes and operations of class. It also shows a collection of classes, interface, association. (Point, n.d.)

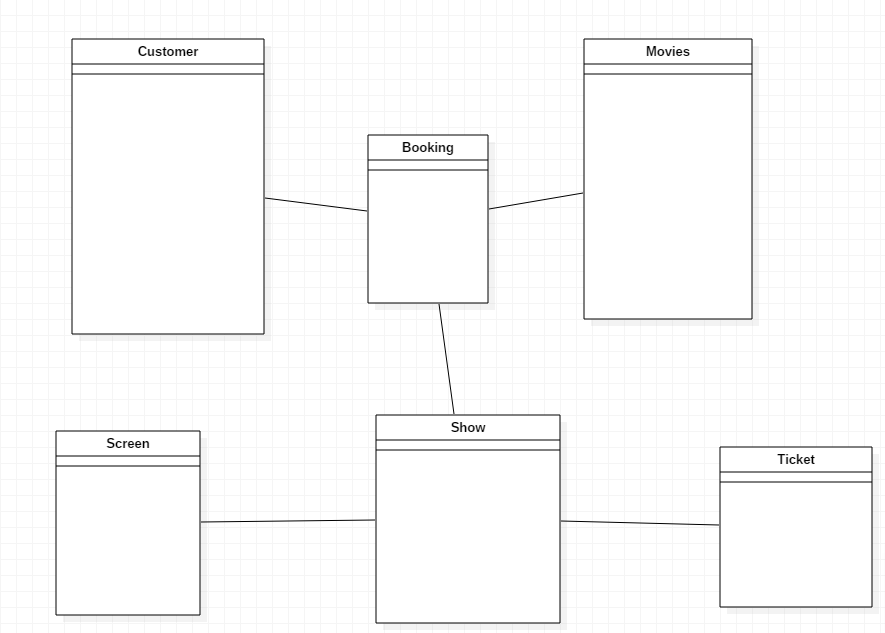


Figure 2 : Initial-class diagram

**3.Database Design**

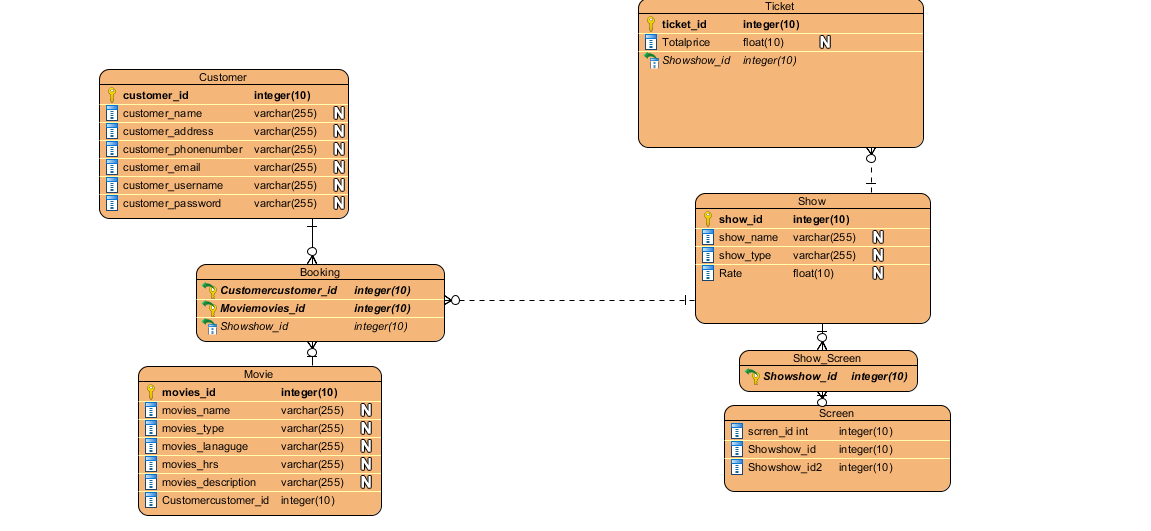


Figure 3: Err-Diagram

**4. UI design Paper Protypes**

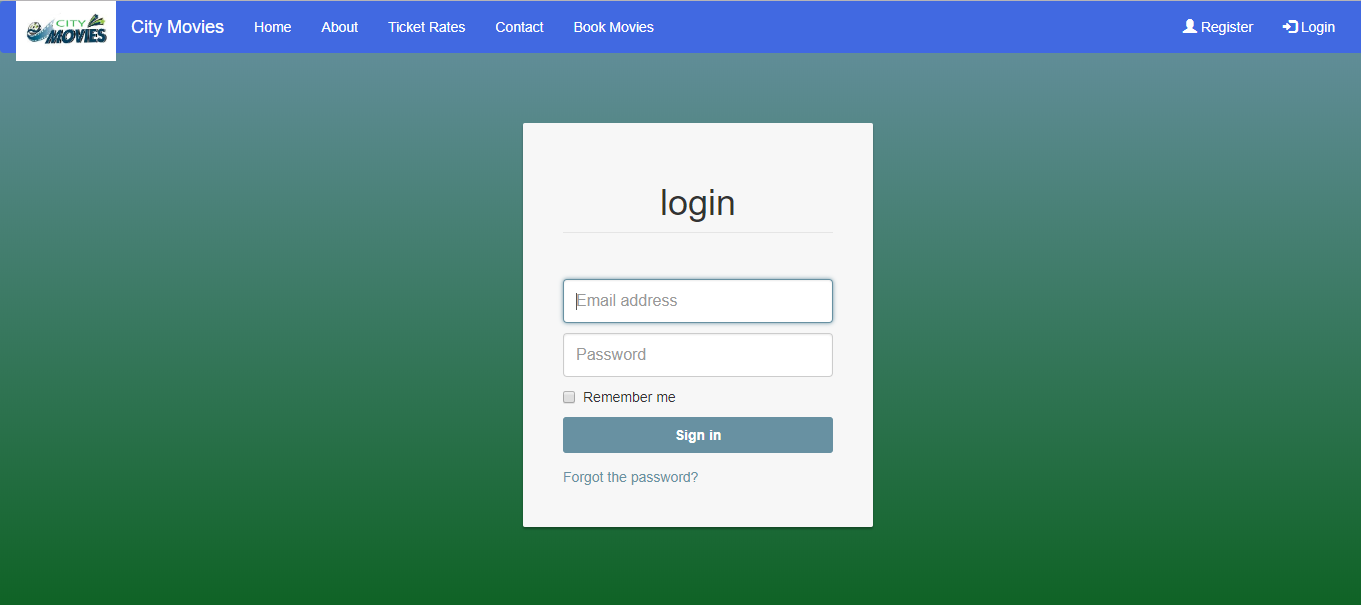


Figure 4: Login form

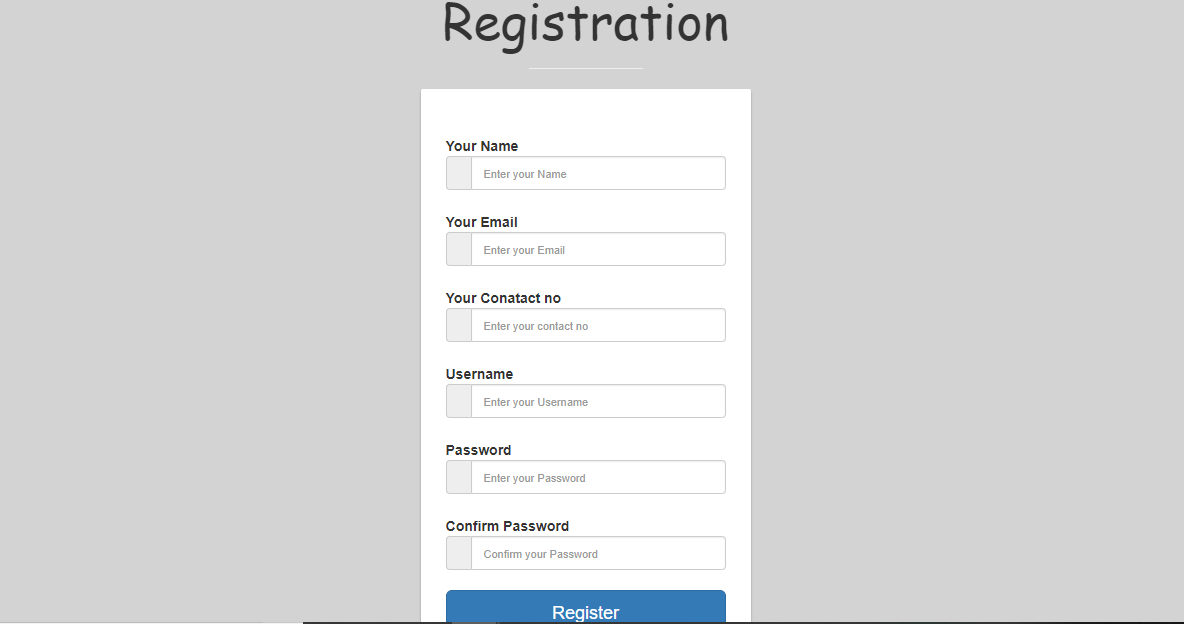


Figure 5: Registration form

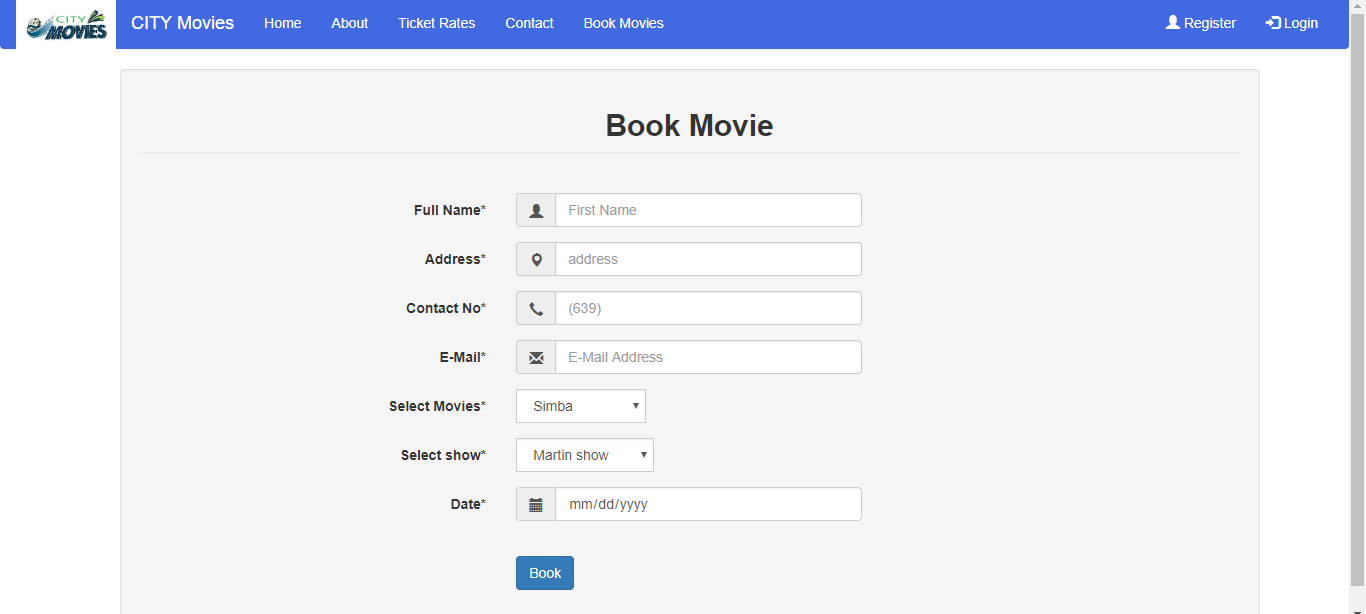


Figure 6: Book Movie

**Conclusion**

Hence, analysis for online movies booking is finally conclude. We find out functional and non-functional requirement for online movie booking system. Use case diagram is drawn to shown the interaction between user and system and initial class diagram also drawn. And finally,database is design for online movies booking system.