**2. Packages**

Large scale projects do not coding by one person, everyone who involves the coding side of project can see very easily, therefore packages are very significant . Before packaging we separate our system to subsystems in our SDD. Also subsystems must be relative with each other, when coding is finish, subsystems attach the code packages. Before the coding process first of all we create empty package, in this package we have necessary statements which is including in our use case. Inside our package user information’s stored. After that our package connect to the DB then retrieval data from DB. Our second package is about creating voyage. Another .php page these information’s taken and stored in DB. In this package we can edit anything payment, booking, see ticket or booking . Finally last package which provide user needs includes Log-in, Sign-up, paying ticket, booking, see ticket ,see booking and main page.

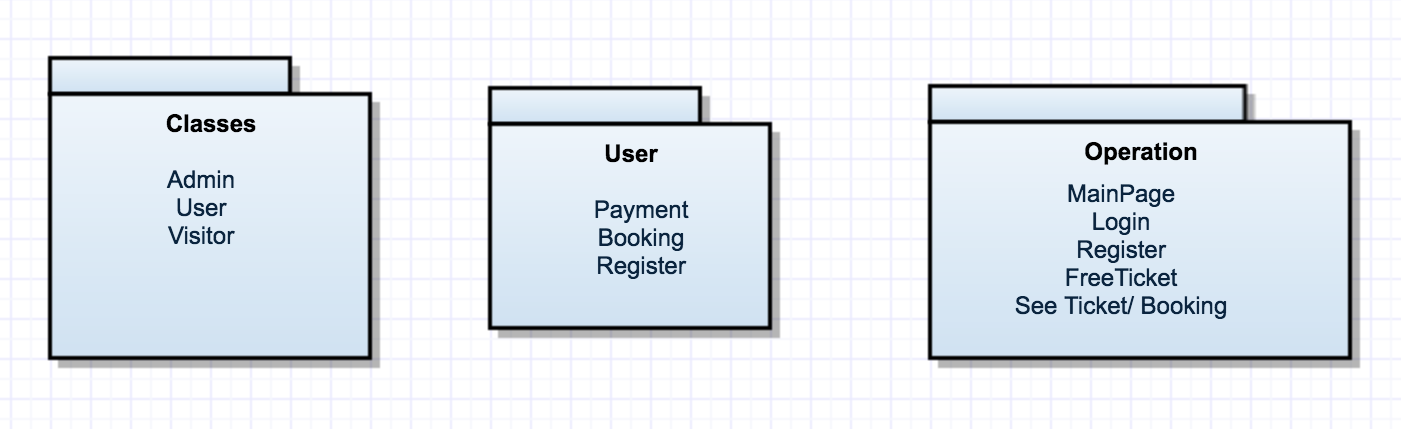


Figure 2.1- OBTS packages

**3. Class Interfaces**

User and Visitor classes are extends from user class. Both of their information are stored in DB and user must be log-in for booking, payment ,ticket and using free ticket and update their information. All classes’ attributes are private. Some necessary part of methods have get and set methods.

**Database**

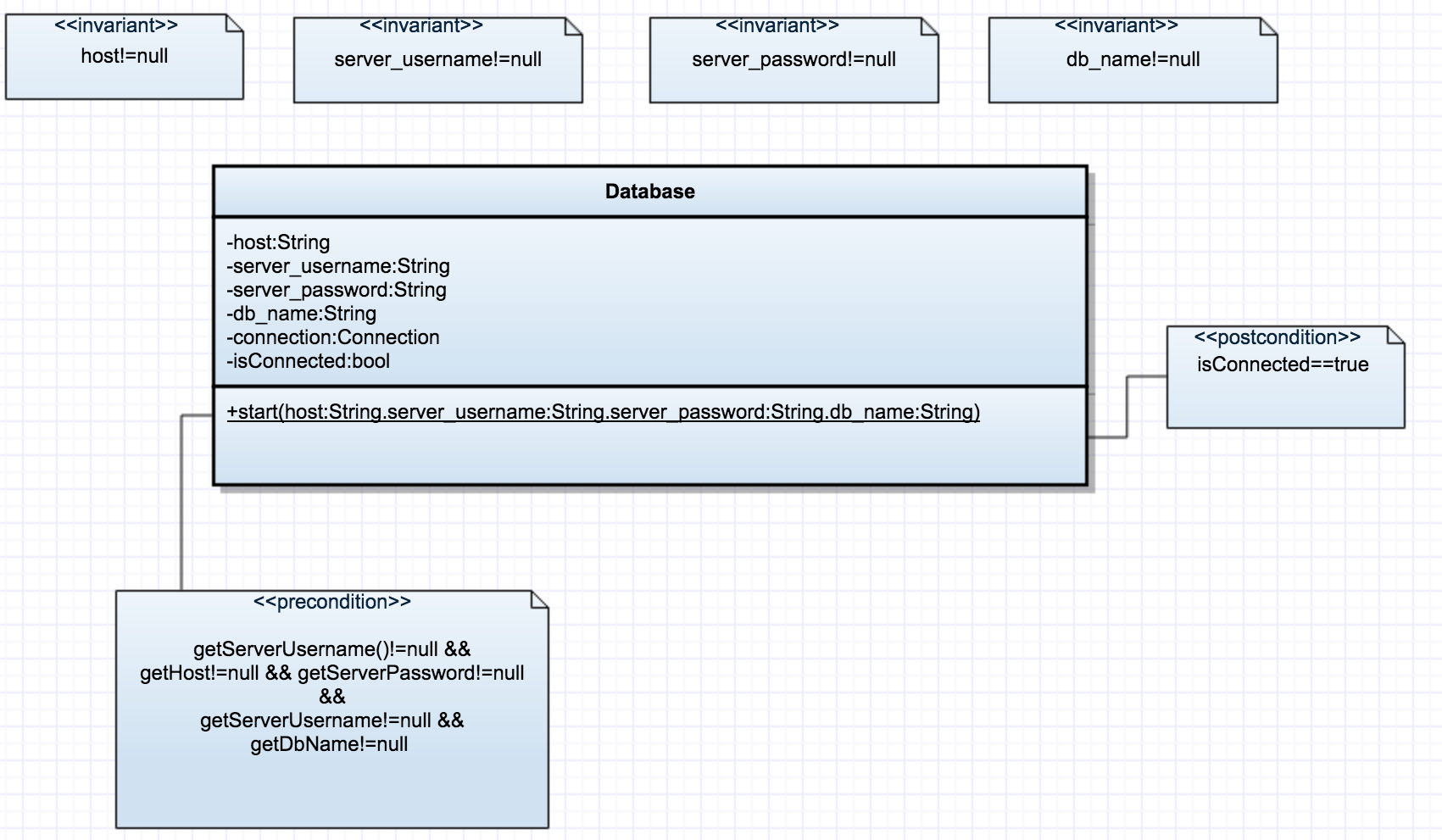
\*Invariants:  
\* These information cannot be null  
\*@invariant db\_name!=null   
\*@invariant server\_password!=null   
\*  @invariant server\_username!=null   
\*  @invariant host!=null \*   
\*  PostConditions:   
\*  start(host,server\_username,server\_password,db\_name)  
\*  @post isConnected==true   
\*PreConditions:  
\* start(host,server\_username,server\_password,db\_name)  
\* @pre getServerUsername()!=null && getHost!=null && getServerPassword!=null && getServerUsername!=null && getDbName!=null \*

Figure 3.1- Database UML

**Voyage**

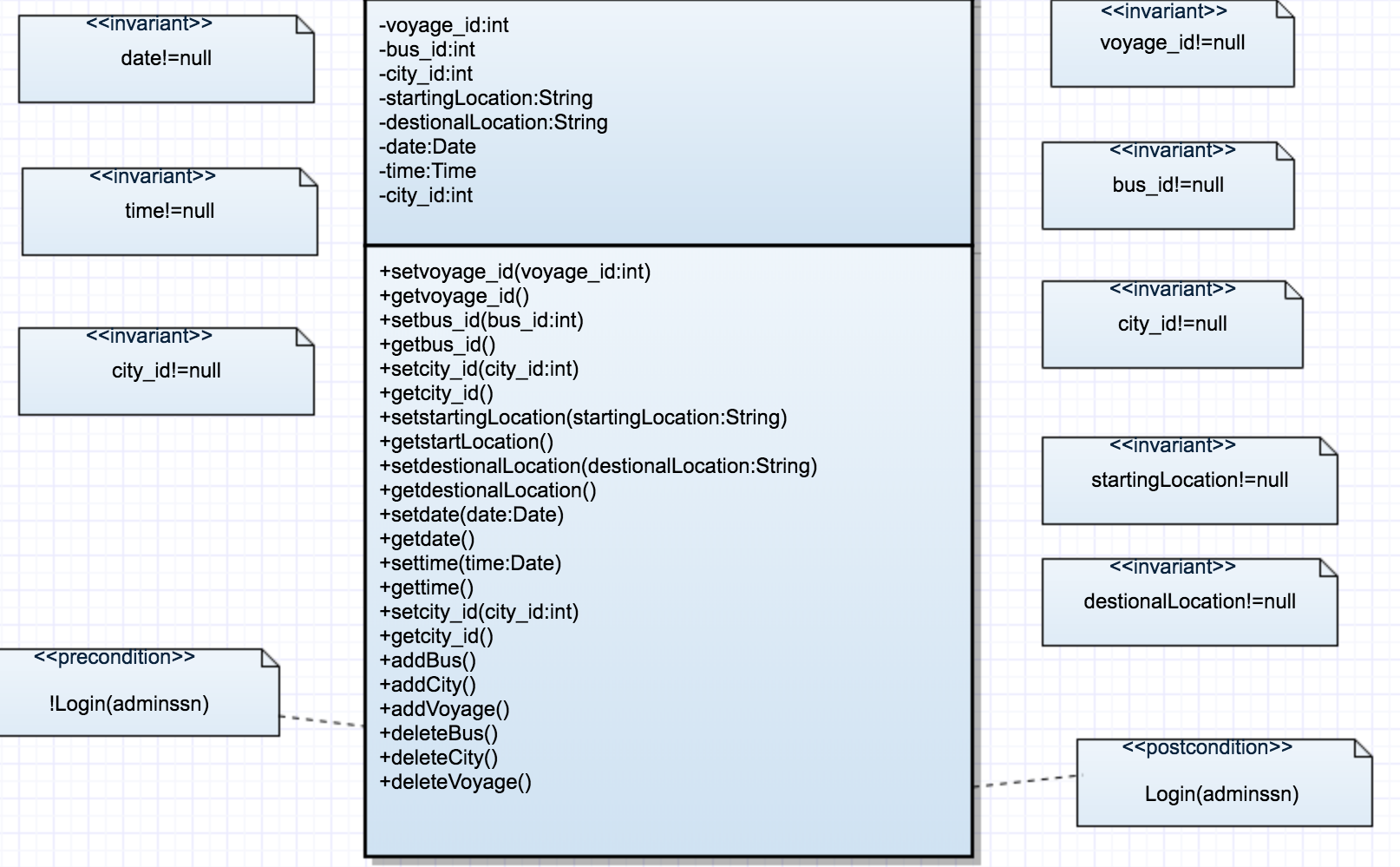
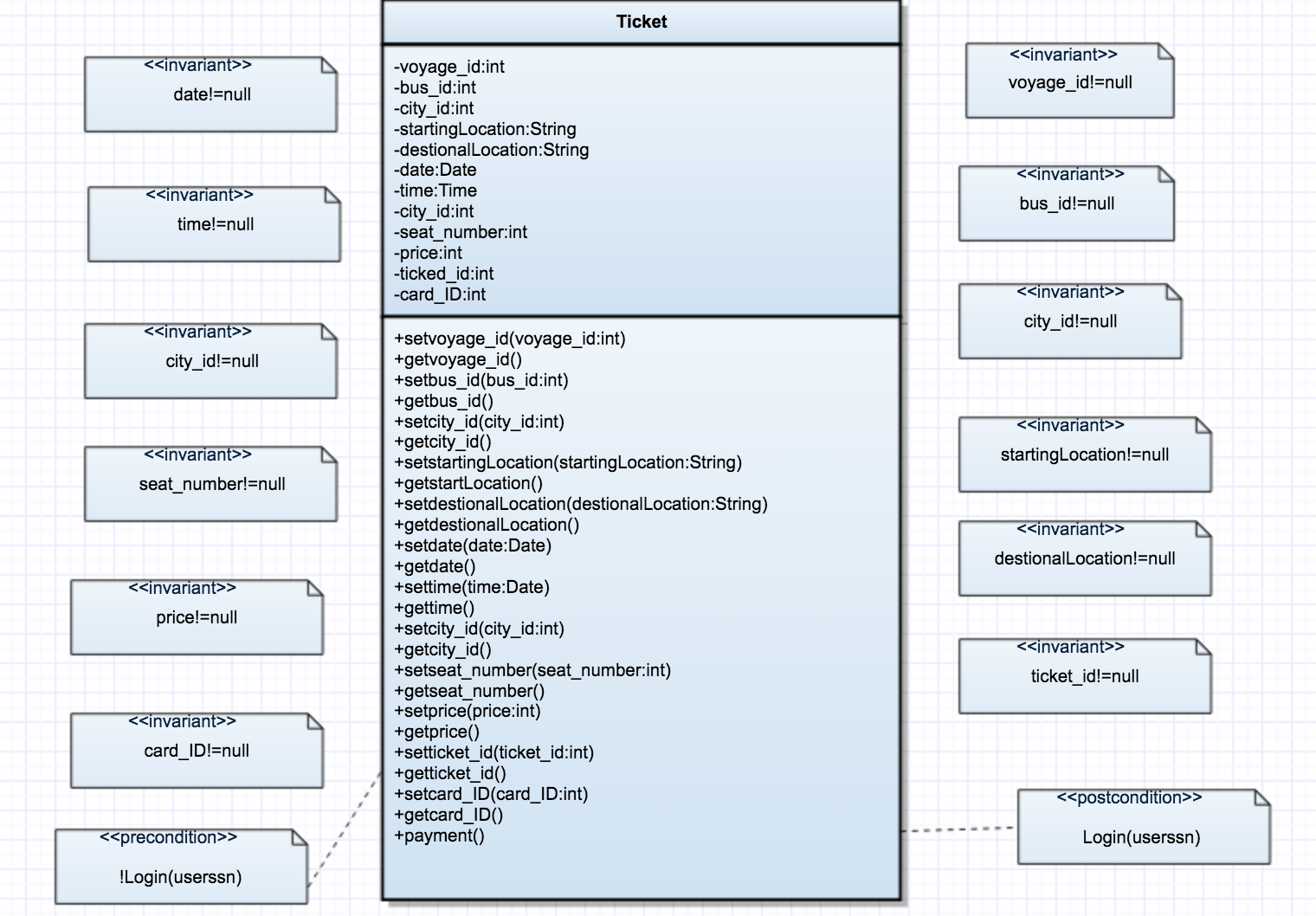
\*Invariants:  
\*@invariant voyage\_id!=null  
\*@invariant bus\_id!=null   
\*@invariant city\_id!=null   
\*@invariant startingLocation!=null   
\*@invariant destionalLocation!=null  
\*@invariant date!=null  
\*@invariant time!=null  
\*   
\*  Preconditions:   
\*  admin\_login(ssn, password)   
\*  @pre getpassword()!=null   
\*  PostConditions:   
\*  isLogin(ssn, password)   
\* @post isLogin == true \*  
\*

Figure 3.2- Voyage UML

**Ticket**

\*Invariants:  
\*@invariant voyage\_id!=null  
\*@invariant bus\_id!=null   
\*@invariant city\_id!=null   
\*@invariant startingLocation!=null   
\*@invariant destionalLocation!=null  
\*@invariant date!=null  
\*@invariant time!=null  
\*@invariant seat\_number!=null  
\*@invariant price!=null  
\*@invariant ticket\_id!=null  
\*@invariant card\_id!=null\*  
\*  Preconditions:   
\*  login(userssn, password)   
\*  @pre getpassword()!=null \*   
\*  PostConditions: \*  Login(userssn, password)   
\*  @post Login == true \*  
Figure 3.3- Ticket UML

Booking

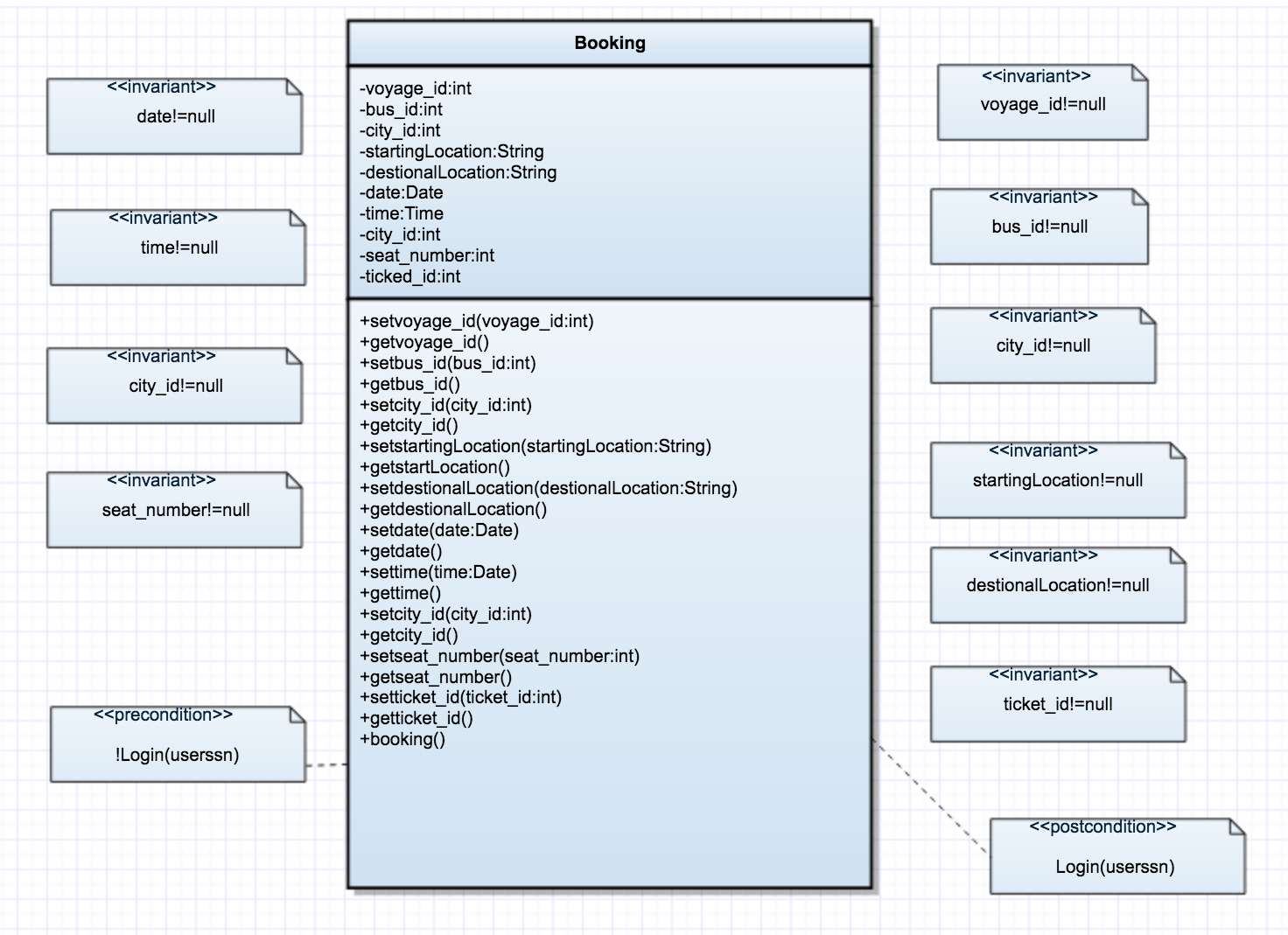
\*Invariants:  
\*@invariant voyage\_id!=null  
\*@invariant bus\_id!=null   
\*@invariant city\_id!=null   
\*@invariant startingLocation!=null   
\*@invariant destionalLocation!=null  
\*@invariant date!=null  
\*@invariant time!=null  
\*@invariant seat\_number!=null  
\*@invariant ticket\_id!=null  
\*  Preconditions:   
\*  login(userssn, password)   
\*  @pre getpassword()!=null \*   
\*  PostConditions: \*  Login(userssn, password)   
\*  @post Login == true \*

Figure 3.4- Booking UML

User

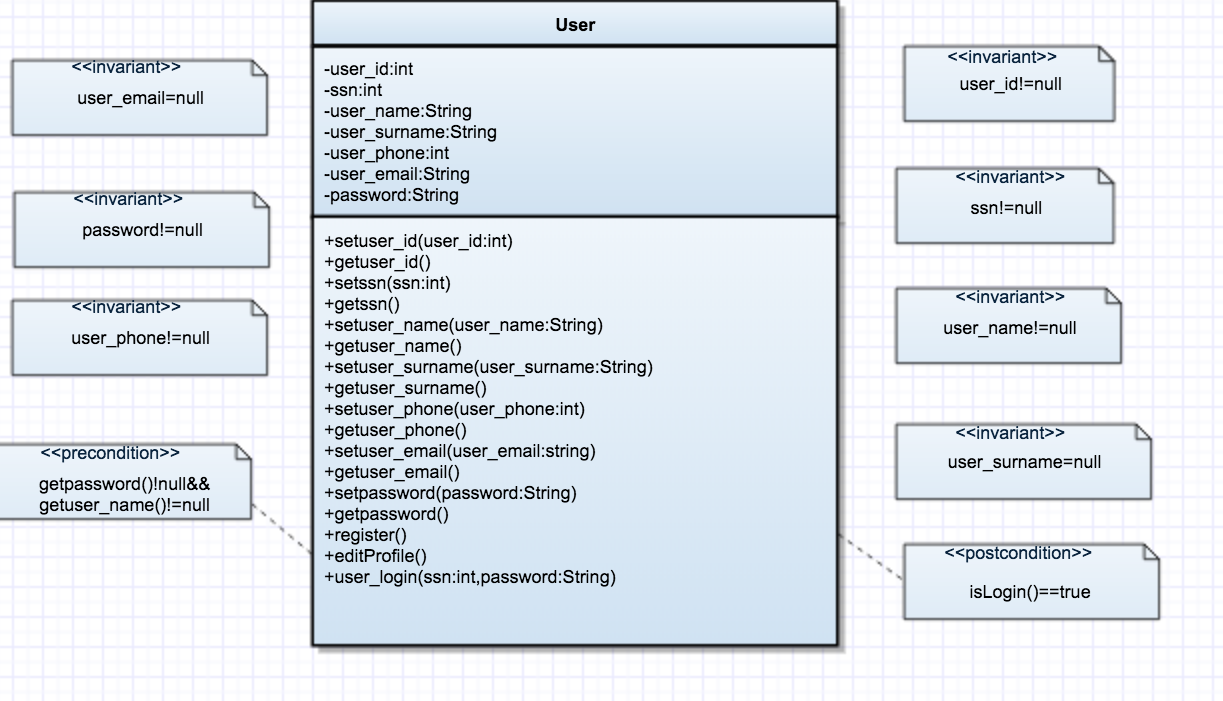
\*@invariant user\_id!=null   
\*@invariant ssn!=null  
\*@invariant user\_name!=null   
\*@invariant password!=null  
 \*@invariant user\_surname!=null  
\*@invariant email!=null  
\*@invariant phone!=null\*   
\*  Preconditions:   
\*  login(userssn, password)   
\*  @pre getpassword()!=null \*   
\*  PostConditions:   
\*  Login(userssn, password)   
\*  @post Login == true \*  


Figure 3.5- User UML