```
Login (Uid, password) {
      u = UserFind (Uid);
      valid = (u! = null);
      if (!valid) {
                  msg("Your account doesn't exist");
      } else {
            valid2 = (u.getPassword() == password);
            if (valid2) {
                  u.setIsLogin(true);
                  u.setLastAction (getCurrentTime());
                  type = UserGetUserType(Uid);
                  return type;
            } else {
                  msg("your password is incorrect");
            }
      }
}
```

```
Logout(Uid) {
          u = UserFind(Uid);
          u.setIsLogin(false);
          msg("successful!");
}
```

```
Kickout (time) {
      UsersVector = UserFindAll();
      for(int i = 0 ; i < UsersVector.size(); i++){
            u = UsersVector.get(i);
            status = (time - u.getLastAction() > 15 min);
            if (status == true) {
                  u.setIsLogin(false);
      }
}
```

```
Reserve (Uid, Sid) {
     u = UserFind (Uid);
     u.setLastAction (getCurrentTime());
     s = ReserveFind(Sid);
     valid = (s == null);
     if (valid) {
          reserveStatus = ReserveAdd(Uid,Sid);
          if(reserveStatus)
               msg ("Successful");
          else
               msg ("Unsuccessful");
     } else
               msg ("This time is full!");
}
```

```
Cancel (Uid, Sid) {
     u = UserFind(Uid);
     u.setLastAction(getCurrentTime());
     scheduleRecord = ScheduleFind(Sid);
     time = (scheduleRecord.getDay()- getCurrentTime() > 1 Day);
     if (time == true) {
          result = ReserveDelete(Sid);
          if (result == true)
               msg ("Cancelation Successful");
          else
               msg ("Unsuccessful");
     } else {
          msg ("Request time out !");
}
```

```
Change (Uid, oldSid, newSid) {
     u = UserFind(Uid);
     u.setLastAction(getCurrentTime());
     r = ReserveFind(oldSid);
     r.setSid(newSid);
     msg("Successful");
}
```

```
Pay (Pid, Did) {
      u = UserFind(Pid);
      u.setLastAction(getCurrentTime());
      price = DoctorGetVisitPrice(Did);
      bankBool = makeBankTransaction(price);
      if (bankBool == true)
            msg (“Successful”);
      else
            msg (“Unsuccessful”);
}


makeBankTransaction (amount) {

      transacted = transact(amount);

      valid = (amount == transacted);

      return valid;

}
```

```
Observe (Pid, Did){
     u = UserFind(Pid);
     u.setLastAction(getCurrentTime());
     scheduleTable = ScheduleShowTableByDoctorId(Did);
     valid = (scheduleTable == null);
     if(valid)
          msg ("The doctor doesn't exist");
     else
          return scheduleTable;
     }
}
```

```
RegisterDoctor (Uid, firstName, lastName, username, password,
                 email, visitPrice, expert) {
     u =UserFindAdmin();
     u.setLastAction(getCurrentTime());
     result = UserInsert(Uid, firstName,
                           lastName, username,
                           password, email,
                           "doctor", true,
                           getCurrentDateTime);
     if (result == true){
           doctorStatus =
                 DoctorInsert(Uid, expert, visitPrice);
           if (doctorStatus == true)
                msg (“Added Successfully”);
           else
                msg (“Unsuccessful”);
     }else
           msg (“User has already exists”);
}
```

```
RegisterPatient (Uid, firstName, LastName, username, password,
                email, age , height , weight){
     u = UserFindAdmin();
     u.setLastAction(getCurrentTime());
     result = UserInsert(Uid, firstName, lastName ,
                         username, password, email ,
                         "patient" , true ,
                         getCurrentTime());
     if(result == false)
          msg ("this user had been added before");
     else{
          patientStatus =
               PatientInsert(Uid, age, height , weight);
          if(patientStatus)
               msg ("Successful");
          else
               msg ("Unsuccessful");
     }
}
```

```
ObservePlan (Uid) {
      u = UserFind(Uid);
      u.setLastAction(getCurrentTime());
      reserveTable = getReserveTable();
      return reserveTable;
}
```

```
CancelDay (Uid, day) {
     u = UserFind(Uid);
     u.setLastAction(getCurrentTime());
     time = (getCurrentTime() - day >= 3 day);
     if(time == true) {
          patientsVector = ReserveFindAllPatient(Uid);
          sendEmailTo(patientsVector);
          reserveStatus = ReserveDelete(Uid);
          scheduleStatus = ScheduleDelete(Uid,day);
          if(reserveSatatus && scheduleStatus)
               msg ("Successful");
          else
               msg ("Unsuccessful");
     }
     else
          msg ("You can't cancel this day");
}
```

```
ObservePatient (Uid){
        u = UserFind(Uid);
        u.setLastAction(getCurrentTime());
        reserveTable = getReserveTableByDoctorID(Uid);
        return reserveTable;
}
```

```
TimeAnnounce (Uid, day, startTime, endTime) {
     u = UserFind(Uid);
     u.setLastAction(getCurrentTime());
     scheduleStatus =
          ScheduleInsert(Uid, date, startTime, endTime);
     if(scheduleStatus == true)
          msg ("Successful");
     else
          msg ("Unsuccessful");
     }
}
```