**Promela code for Login Sequence diagram:**

mtype = {login, insert\_email\_pass, inserted\_email\_pass, verify\_email\_pass, verified\_email\_pass, unverified\_email\_pass,

         login\_success, invalid\_user};

chan user = [2] of {mtype, bit};

chan system = [2] of {mtype, bit};

chan database = [2] of {mtype, bit};

bool verify = 1;

bool valid = 1;

bool verified = 1;

proctype User(chan userCh, systemCh, databaseCh)

{

    bit snd,rcv;

    do

    :: userCh!login(snd)->systemCh?insert\_email\_pass(rcv);

    userCh!inserted\_email\_pass(snd);

    if

    ::valid==1->

        userCh?login\_success(rcv);valid=0;

    ::valid==0->

        userCh?invalid\_user(rcv);valid=1;

    fi

    od

}

proctype System(chan userCh, systemCh, databaseCh)

{

    bit snd,rcv;

    do

    ::userCh?login(rcv)->systemCh!insert\_email\_pass(rcv);

    userCh?inserted\_email\_pass(rcv)->systemCh!verify\_email\_pass(rcv);

    if

    ::verified==1->

        databaseCh?verified\_email\_pass(rcv)->userCh!login\_success(rcv);verified=0;

    ::verified==0->

        databaseCh?unverified\_email\_pass(rcv)->userCh!invalid\_user(rcv);verified=1;

    fi

    od

}

proctype Database(chan userCh, systemCh, databaseCh)

{

    bit rcv;

    do

    ::if

    ::verify==1->

        system?verify\_email\_pass(rcv)->databaseCh!verified\_email\_pass(rcv);verify=0;

    ::verify==0->

        system?verify\_email\_pass(rcv)->databaseCh!unverified\_email\_pass(rcv);verify=1;

    fi

    od

}

init

{

run User(user, system, database);

run System(user, system, database);

run Database(user, system, database);

}

**Promela code for Covid News Update Sequence diagram:**

mtype = {upWoP\_req, upWoP\_ack, upWoRe\_req, upWoRe\_ack, upWoDe\_req, upWoDe\_ack,

upBdP\_req, upBdP\_ack, upBdRe\_req, upBdRe\_ack, upBdDe\_req, upBdDe\_ack,

     shoWoP\_ack, shoWoRe\_ack, shoWoDe\_ack, shoBdP\_ack, shoBdRe\_ack, shoBdDe\_ack,

shoWoP, shoWoRe, shoWoDe, shoBdP, shoBdRe, shoBdDe};

chan user = [2] of {mtype, bit};

chan system = [2] of {mtype, bit};

chan database = [2] of {mtype, bit};

proctype System(chan userCh, systemCh, databaseCh)

{

    bit snd,rcv;

    do

    ::systemCh!upWoP\_req(snd)->databaseCh?upWoP\_ack(rcv);

    systemCh!upWoRe\_req(snd)->databaseCh?upWoRe\_ack(rcv);

    systemCh!upWoDe\_req(snd)->databaseCh?upWoDe\_ack(rcv);

    systemCh!upBdP\_req(snd)->databaseCh?upBdP\_ack(rcv);

    systemCh!upBdRe\_req(snd)->databaseCh?upBdRe\_ack(rcv);

    systemCh!upBdDe\_req(snd)->databaseCh?upBdDe\_ack(rcv);

    systemCh!shoWoP(snd)->userCh?shoWoP\_ack(rcv);

    systemCh!shoWoRe(snd)->userCh?shoWoRe\_ack(rcv);

    systemCh!shoWoDe(snd)->userCh?shoWoDe\_ack(rcv);

    systemCh!shoBdP(snd)->userCh?shoBdP\_ack(rcv);

    systemCh!shoBdRe(snd)->userCh?shoBdRe\_ack(rcv);

    systemCh!shoBdDe(snd)->userCh?shoBdDe\_ack(rcv);

    od

}

proctype Database(chan userCh, systemCh, databaseCh)

{

    bit rcv;

    do

    ::systemCh?upWoP\_req(rcv)->databaseCh!upWoP\_ack(rcv);

    systemCh?upWoRe\_req(rcv)->databaseCh!upWoRe\_ack(rcv);

    systemCh?upWoDe\_req(rcv)->databaseCh!upWoDe\_ack(rcv);

    systemCh?upBdP\_req(rcv)->databaseCh!upBdP\_ack(rcv);

    systemCh?upBdRe\_req(rcv)->databaseCh!upBdRe\_ack(rcv);

    systemCh?upBdDe\_req(rcv)->databaseCh!upBdDe\_ack(rcv);

    od

}

proctype User(chan userCh, systemCh, databaseCh)

{

    bit rcv;

    do

    ::systemCh?shoWoP(rcv)->userCh!shoWoP\_ack(rcv);

    systemCh?shoWoRe(rcv)->userCh!shoWoRe\_ack(rcv);

    systemCh?shoWoDe(rcv)->userCh!shoWoDe\_ack(rcv);

    systemCh?shoBdP(rcv)->userCh!shoBdP\_ack(rcv);

    systemCh?shoBdRe(rcv)->userCh!shoBdRe\_ack(rcv);

    systemCh?shoBdDe(rcv)->userCh!shoBdDe\_ack(rcv);

    od

}

init

{

run System(user, system, database);

run Database(user, system, database);

run User(user, system, database);

}

**Promela code for Test Request Sequence diagram:**

mtype = {req\_for\_corona\_test, select\_test\_center, selected\_test\_center,

select\_date, selected\_date, store\_data, data\_stored,

request\_successful, ack};

chan user = [2] of {mtype, bit};

chan system = [2] of {mtype, bit};

chan database = [2] of {mtype, bit};

proctype User(chan userCh, systemCh, databaseCh)

{

    bit snd,rcv;

    do

    :: userCh!req\_for\_corona\_test(snd)->systemCh?select\_test\_center(rcv);

    userCh!selected\_test\_center(snd)->systemCh?select\_date(rcv);

    userCh!selected\_date(snd)->systemCh?request\_successful(rcv);

    od

}

proctype System(chan userCh, systemCh, databaseCh)

{

    bit snd,rcv;

    do

    ::userCh?req\_for\_corona\_test(rcv)->systemCh!select\_test\_center(rcv);

    userCh?selected\_test\_center(rcv)->systemCh!select\_date(rcv);

    userCh?selected\_date(rcv)->systemCh!store\_data(rcv);

    databaseCh?data\_stored(snd)->systemCh!request\_successful(rcv);

    od

}

proctype Database(chan userCh, systemCh, databaseCh)

{

    bit rcv;

    do

    ::systemCh?store\_data(rcv)->databaseCh!data\_stored(rcv);

    od

}

init

{

run User(user, system, database);

run System(user, system, database);

run Database(user, system, database);

}