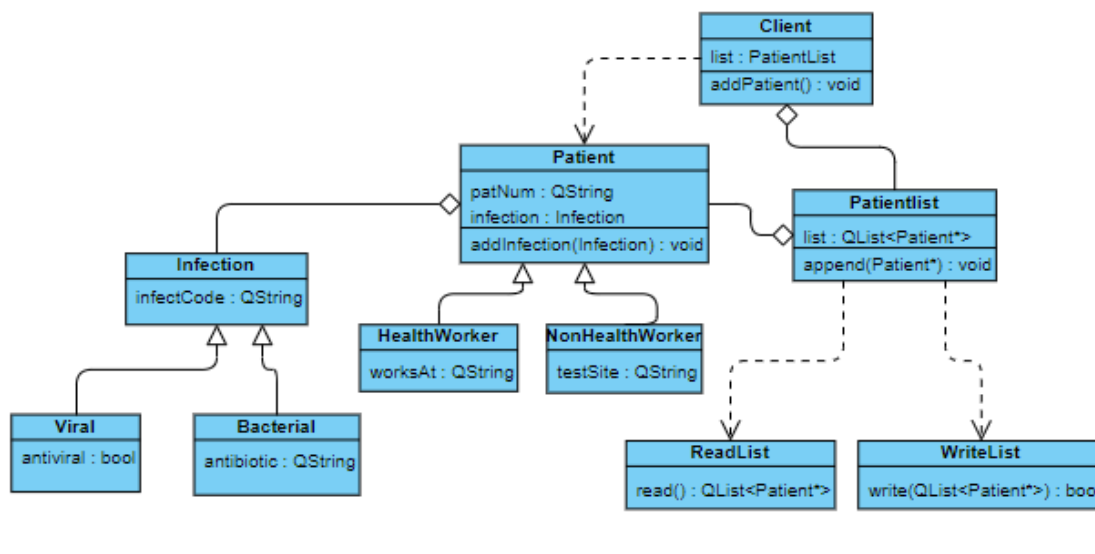


Question 1

1.1



1.2

1.2.1 Input Kludge

1.2.2

```
QRegExp reg1("[1][1][0-9]{2} (f | m)?[A-Z]{1}[a-zA-Z]{2,}");
```

```
QRegExpValidator *reValid(new QRegExpValidator(reg1, this));
```

```
inputPatientNumber->setValidator(reValid);
```

1.2.3 No , first digit must be "1" .

1.2.4 Yes, 1005 is Intermediate and can be entered into QLineEdit .

1.2.5 No , Input mask cannot handle requirement of "1" as first character . nor optional gender character "f or m" in the patient number.

Question 2

2.1 Although a base class pointer can point to objects of derived classes, polymorphism (dynamic/runtime binding) is not implemented because `toString()` is not virtual in base class `Bed`. Base class `toString()` method is called, not derived class `toString()` method .

2.2

```
QDomDocument writeDom(Bed *object)
{
    QDomDocument doc ;

    cont QMetaObject *mo = object->metaobject();

    QDomElement bedElement = doc.createElement("bed");
    bedElement.setAttribute("type", mo->className);
    doc.appendChild(bedElement);

    for(int i = 1 , i < mo->propertyCount() , i++)
    {
        cont QMetaProperty prop = mo->property(i);
        QDomElement element = doc.createElement(prop.name());
        bedElement.appendChild(element);

        QString value = object->property(prop.name()).toString();
        QDomText text = doc.createTextNode(value);
        element.appendChild(text);
    }
    return doc;
}
```

2.3 No difference when using meta-object.

getNumber() not accessible outside the class.

2.4 Bed bed

```
bed.setProperty("NOTE", "Isolation");
```

Question 3

3.1

Drawback – QAbstractTableModel is ABSTRACT , and cannot be used directly . Must be sub-classed , and methods implemented.

Benefit – Flexibility – Can be sub-classed to provide a suitable table model to display 2 dimensional data for the bed data .

3.2 Model must be linked to a suitable view(QTableView) – view->setModel(model) ;

Need to use inheritance to sub-class a suitable model .

3.3

- Yes , QSIM can be used for table data.
- Some columns may need to be blank – not all beds have same data.
- It is possible to place data into specific columns / cells .

3.4 A delegate can be used to customize the view , and display check boxes, or check state, for both viewing and editing the data

3.5 Only one model to ensure that there is only one “list” of bed states, and that there is no double/multiple booking of beds.

3.6

class BedsSingleton

```
{  
    public :  
        static BedsSingleton* getModel() ;  
  
    private :  
        BedsSingleton() ;  
        static bedsSingleton* singletonInstance ;  
        QStandardItemModel beds ;  
};
```

3.7

```
QProcess *process(new QProcess);  
QStringList arguments ;  
arguments << "vacant " << "vacant beds.txt" ;  
process->start("printBedsToFile" , arguments)
```

```
void QProcess::start(const QString &program, const QStringList &arguments, QIODevice::OpenMode  
mode = ReadWrite)
```

3.8

```
class BedMemento  
{  
    public :  
    private :  
        friend class BedList ;  
        BedMemento() ;  
        QMultiMap<QString, Bed*> getState() ;  
        void setState(QMultiMap<QString , Bed*> s );  
        QMultiMap<QString, Bed*> state ;  
};
```