附件：参考作业报告模板

华南理工大学

South China University of Technology

《软件工程课程设计》作业报告

作业名称：Dental Practice System

学　院　计算机科学与工程学院

专　业　计算机科学与技术

姓　名　黄 冠

任课教师　李 剑

提交日期　2012 年 6 月 25 日

Part 1. Requirement

Exercise 1.1

Indentify requirements from different viewpoints such as patients, the dental assistant, the dental hygienist and the receptionist.

1. Listed in the form of items.

Solution

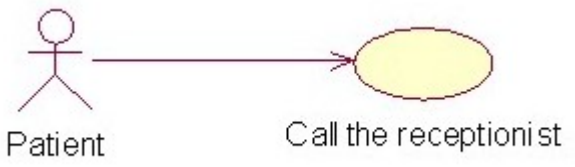
Patients	1. The patient should know the phone number of the dental place. 2. The patient make & cancel appointment via the phone call and the receptionist operate with the computer to make & cancel the appointment.
Receptionist	1. The system should has a login system. Users log in the dental system before doing other things. 2. The system should save the patients' information for next time revisiting. 3. The user can make & cancel appointments in a calendar. 4. The system can print out a notification list for making reminder calls 2 days before appointments. 5. The system print out daily and weekly work schedules with all the patients.
Dental hygienist & dental assistant	1. The system can mark the appointment as completed and add comments. 2. The system can schedule the patient for the next visit. 3. The system can answer queries for patient records by patient name and date.

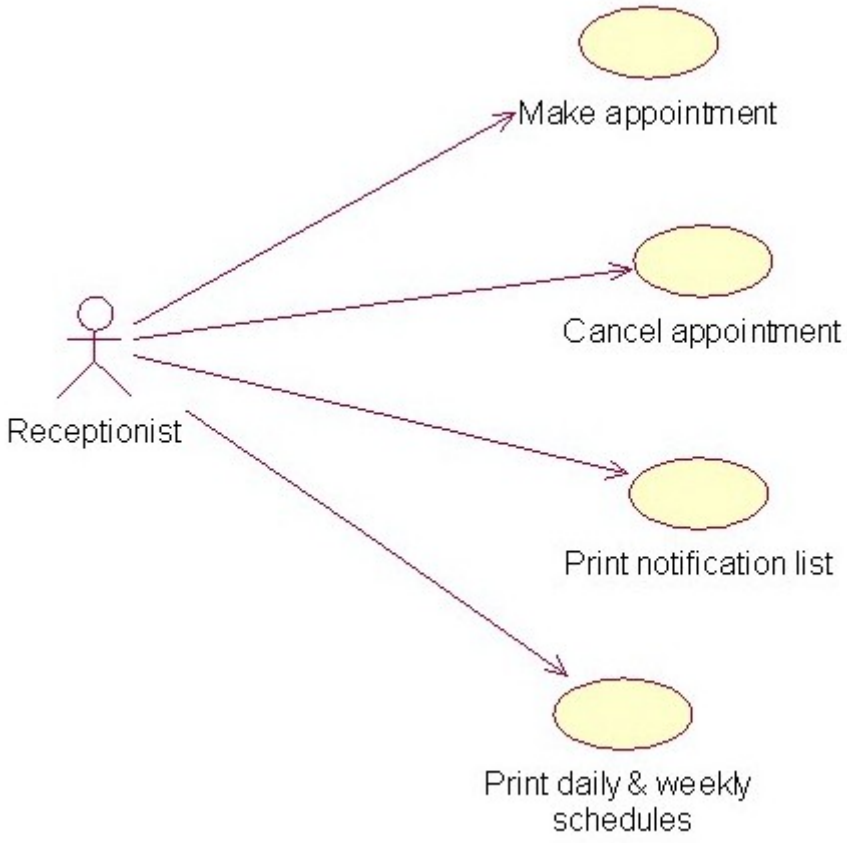
Exercise 1.2

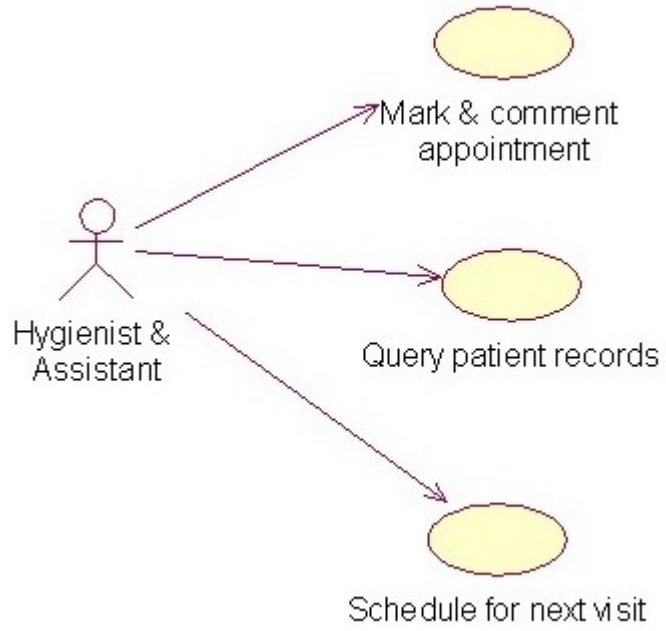
Draw a use case diagram for the dental practice system.

1. To show the dental practice system’s functions and how it interacts with external users.
2. Should be drawn in the form of “Use case diagram” in Rational Rose.

Solution




Make appointment
Cancel appointment
Receptionist
Print notification list
Print daily & weekly schedules


Mark & comment appointment
Hygienist & Assistant
Query patient records
Schedule for next visit

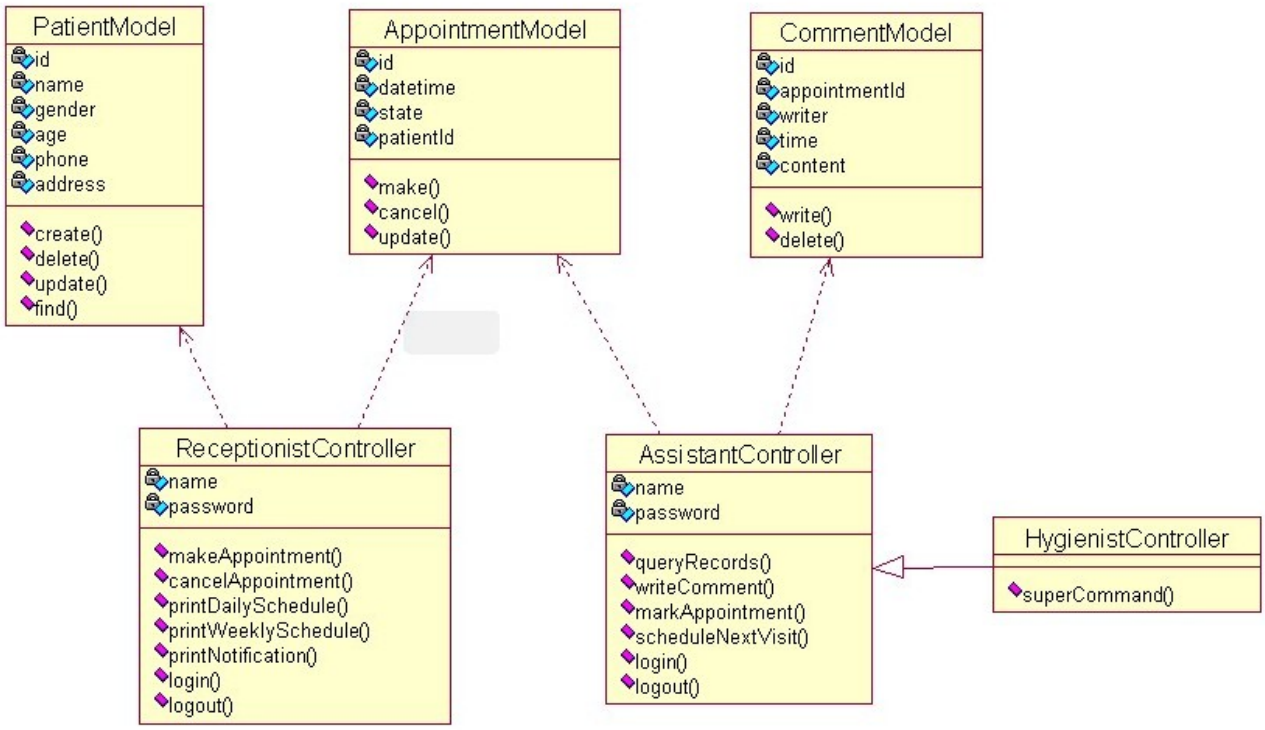
Part 2 Design

Exercise 2.1

Draw a class diagram for the dental practice system.

1. Identify the main classes in the dental practice system.
2. Should be drawn in the form of “class diagram” in Rational Rose.

Solution

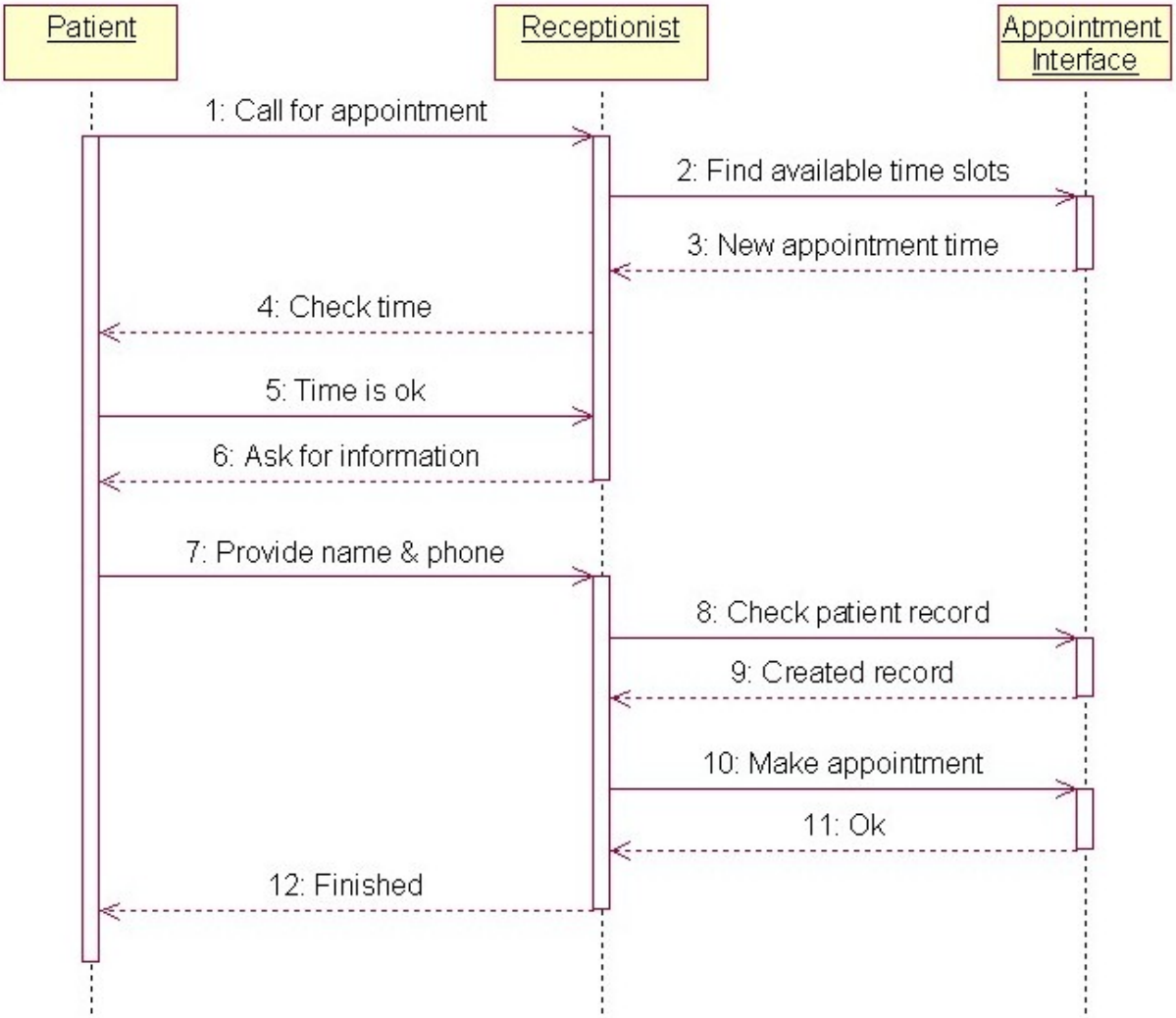


Exercise 2.2

Draw a sequence diagram for the dental practice.

1. To show the process of ‘appointment’.
2. Should be drawn in the form of “sequence diagram” in Rational Rose.

Solution



Part 3. Development

Exercise 3.1

Develop a prototype to implement the function of ‘appointment’.

1. Include user interfaces and the function of ‘appointment’.
2. Implemented in Java or other object-oriented programming languages (Implement the classes in your design).
3. Follow MVC model.

Solution

My prototype is written in C++ with Qt4 (a very popular and portable UI library). I have implemented the 'appointment' function and it approximately follows the following MVC design model.

The model classes	PatientModel, AppointmentModel, CommentModel
The controller classes	AppointmentController
The viewer classes	AppointmentCalendarView, DailyWorkScheduleView, WeeklyWorkScheduleView, NotificationListView, PatientSearchView

The *PatientModel* class:

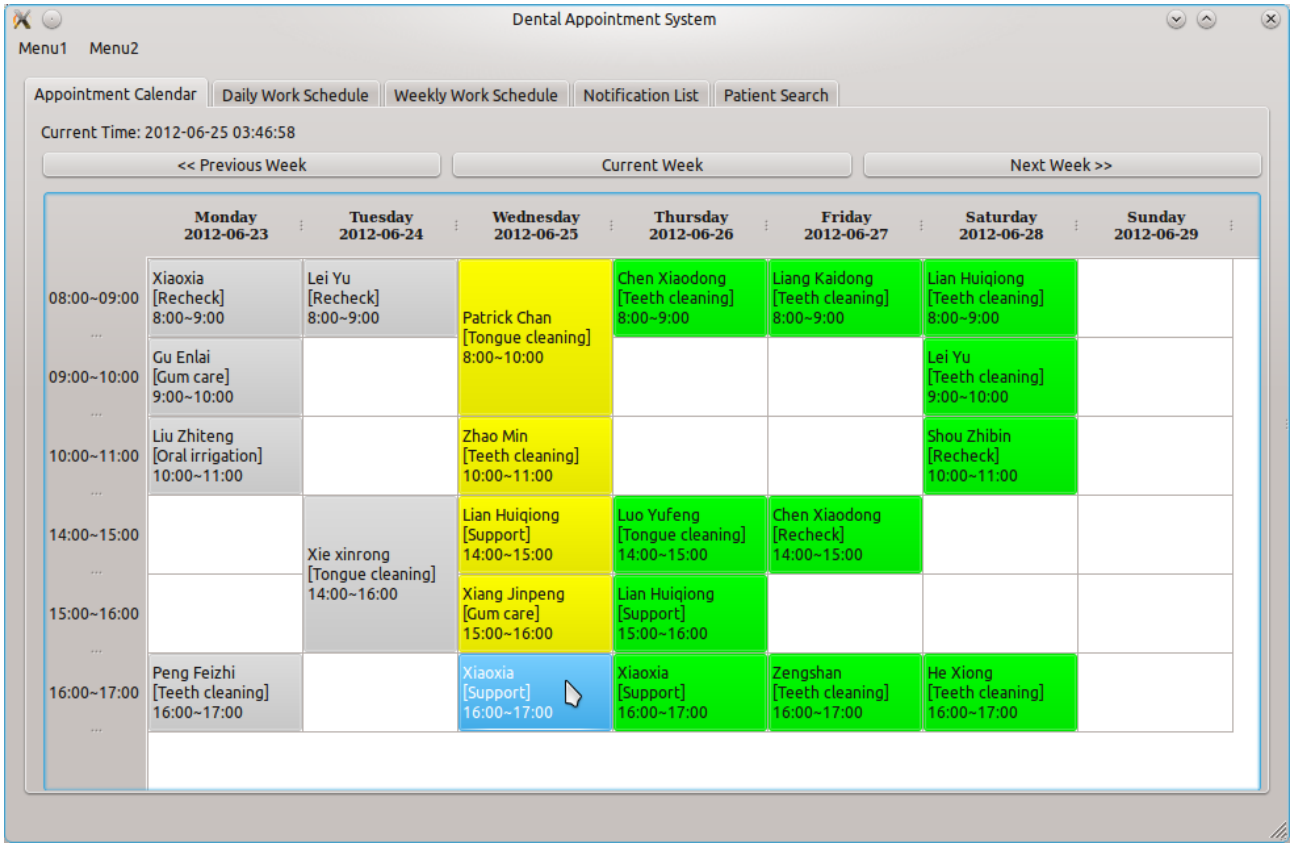
```
class PatientModel : public Model
{
public:
  int id;
  int gender;
  QString name;
  QString phone;
  QString address;
public:
  PatientModel();
};
```

The *AppointmentModel* class:

```
class AppointmentModel : public Model
{
public:
  AppointmentModel();
public:
  int id;
  int patientId;
  QString phone;
  QString purpose;
  QString date;
  int beginTime;
  int endTime;
  QString state;
};
```

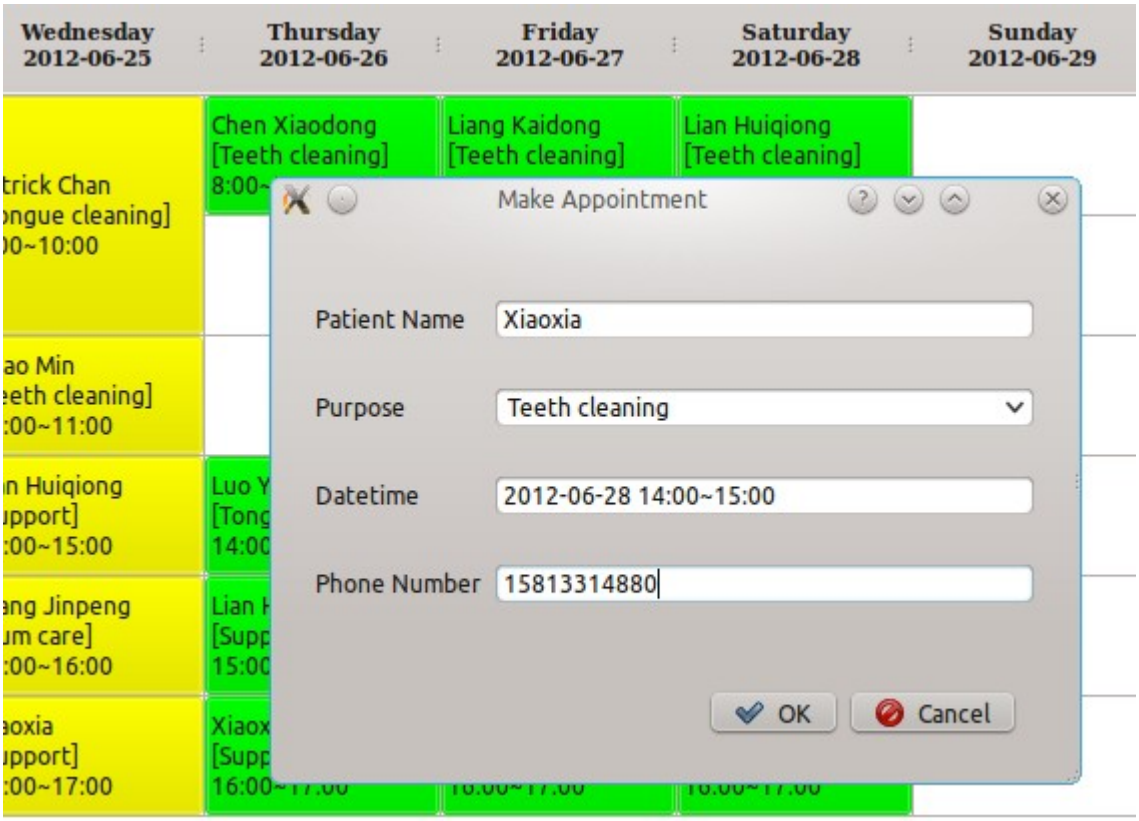
To make an appointment, the receptionist will use the *AppointmentCalendarView*.

The *AppointmentCalendarView* preview:

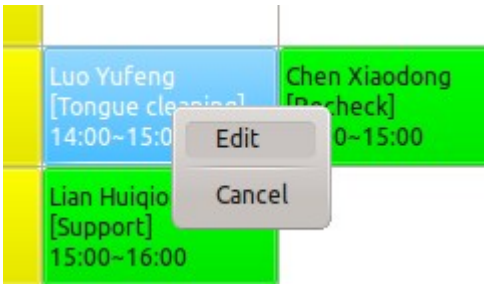


In the preview screenshot, it shows a calendar for scheduling appointments. The receptionist will use this calendar to make appointments with patients. The gray cells mean the past appointments, the yellow highlights the appointments of today while the green ones are the future appointments. Every cell simply displays the name of the patient, the purpose and time interval of the appointment.

The user can right click a blank cell item on the calendar and choose the menu button “Make Appointment”, an AddAppointment Dialog will pop up and ask for information.



If the user click an existing cell item on the calendar, the “Edit” and “Cancel” menu buttons are shown to make changes to the existing appointment. So making two appointments in an intersected time region is impossible.



The *AppointmentController* class:

```
class AppointmentController : public Controller
{
private:
  QList<AppointmentModel*> appointmentList;
public:
  AppointmentController();
  ~AppointmentController();

  AppointmentModel* make(QString name, QString phone, QString purpose, QString date, int
beginTime, int endTime);
  void cancel(int id);
  void mark(int id, int state);
  void comment(int id, QString com);
  AppointmentModel* find(QString date, int beginTime);
  QList<AppointmentModel*> find(QString date);
  QList<AppointmentModel*> find(int patientId);
  QList<AppointmentModel*> findAll();

  void save();
};
```

When the user clicks the “Make Appointment” menu button, the following program will be executed.

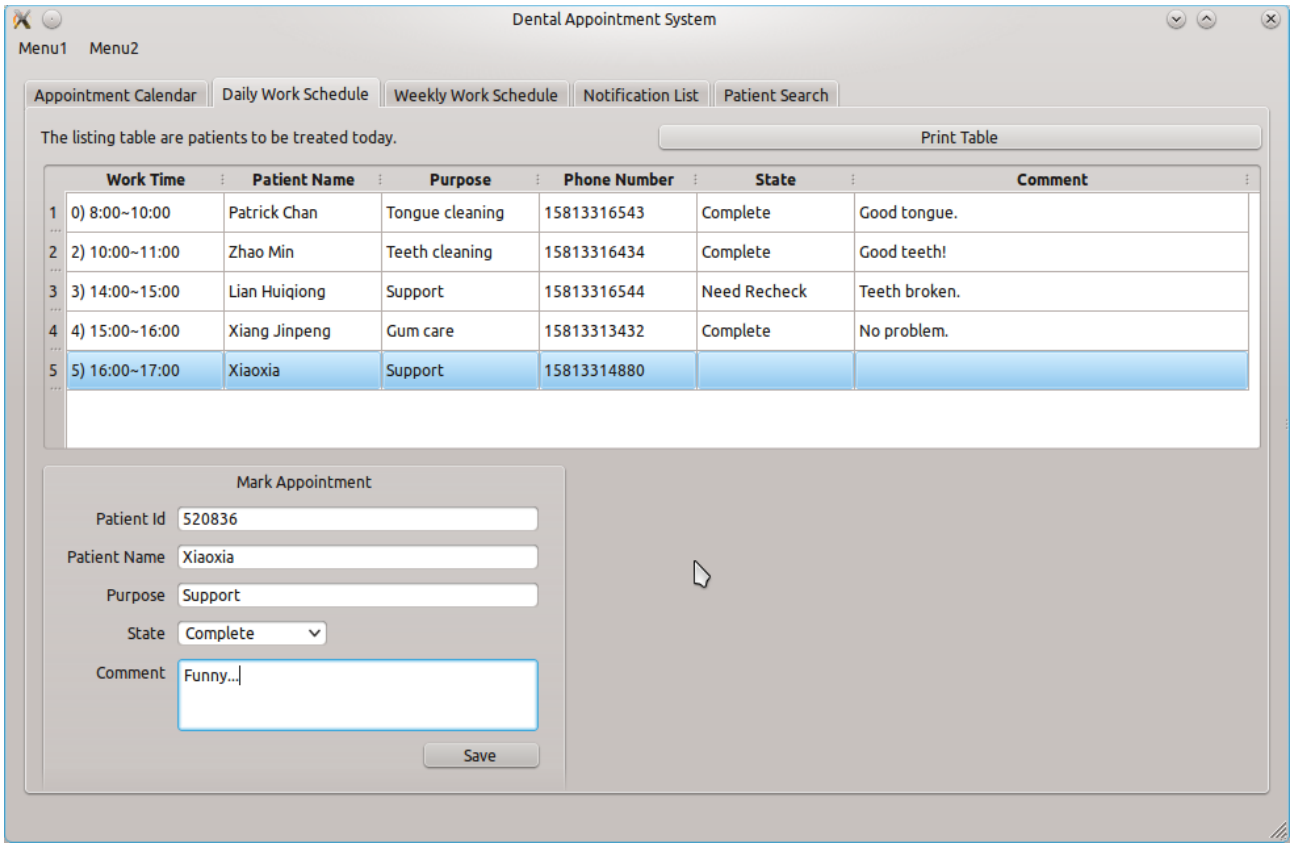
```
void AppointmentCalendarView::on_actionMake_triggered()
{
  QList<QTableWidgetSelectionRange> list = this->ui->calendarView->selectedRanges();
  if(list.size() == 0)
    return;
  QTableWidgetSelectionRange r = list[0];
  /* 禁止选择多列，但可以选择多行，表示预约时间比较长。 */
  if(r.leftColumn() != r.rightColumn()){
    QMessageBox::warning(0, "Invalid", "Multiple columns are not allowed.");
    return;
  }
  /* 计算光标所在的预约的日期 */
  QString date = this->dayOfWeek.addDays(r.leftColumn()).toString("yyyy-MM-dd");
  /* 添加预约信息的对话框 */
  AddAppointment add;
  add.show();
  if(add.exec()){
    if(add.getName().isEmpty()){
      QMessageBox::warning(0, "Invalid", "You must input a valid patient name.");
      return;
    }
    /* 调用 appointmentController 的方法把信息输入到数据库 */
    this->appointmentController.make(add.getName(), add.getPhone(), add.getPurpose(),
date, r.topRow(), r.bottomRow());
    /* 更新 CalendarView */
    this->updateCalendarView();
  }
}
```

The make method of *AppointmentController* creates a new appointment and inserts it into the database.

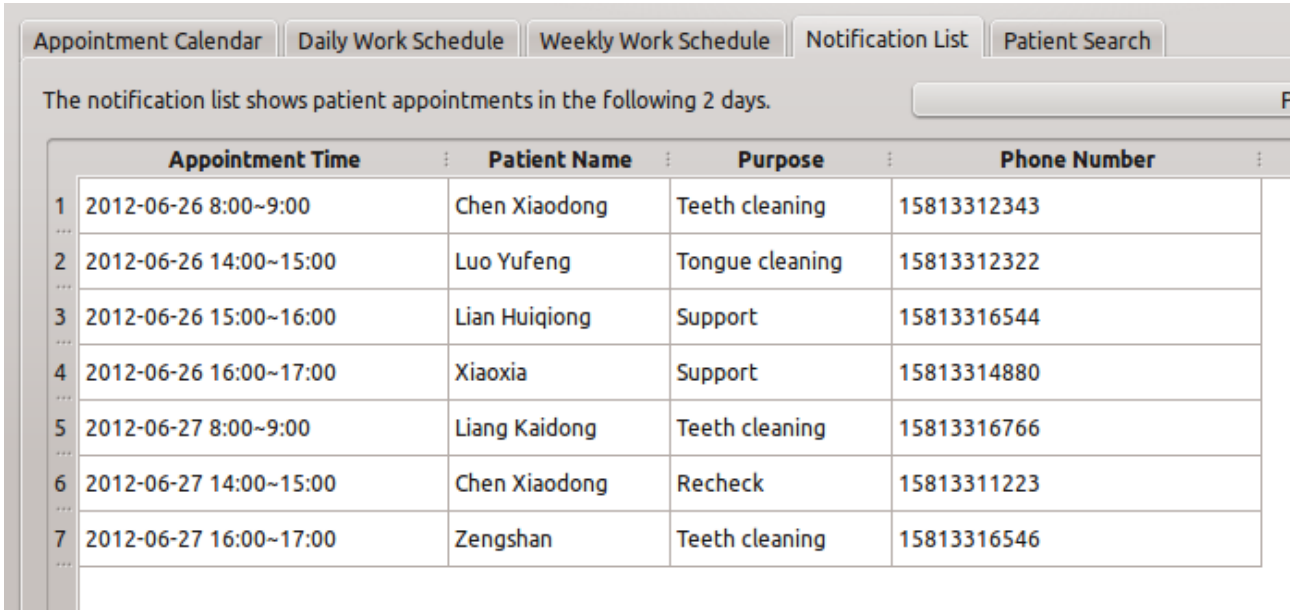
```
AppointmentModel* AppointmentController::make(QString name, QString phone, QString
purpose, QString date, int beginTime, int endTime)
{
  PatientModel* p = new PatientModel(name);
  AppointmentModel* a = new AppointmentModel;
  a->patientId = p->id;
  a->phone = phone;
  a->date = date;
  a->beginTime = beginTime;
  a->endTime = endTime;
  a->purpose = purpose;
  a->id = time(0)%1000000;
  this->appointmentList.append(a);

  printf("Added new appointment %d.\n", a->id);
  return a;
}
```

To mark the appointment as completed and add comments, the assistant or hygienist will use the *DailyWorkScheduleView*.

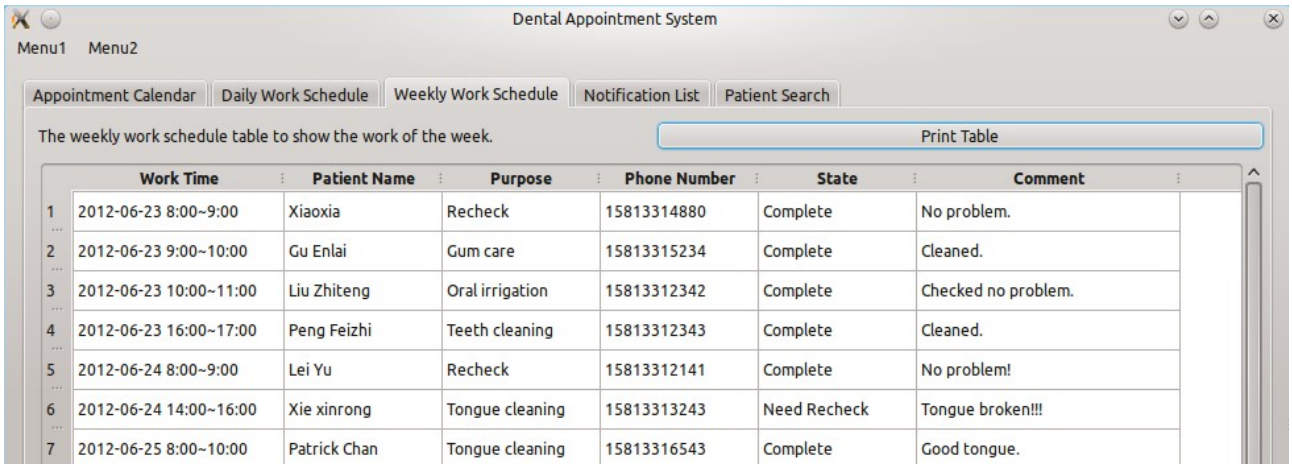


The receptionist might use the *NotificationListView* to gain a list of patients that will have appointments in 2 days and remind them as soon as possible.



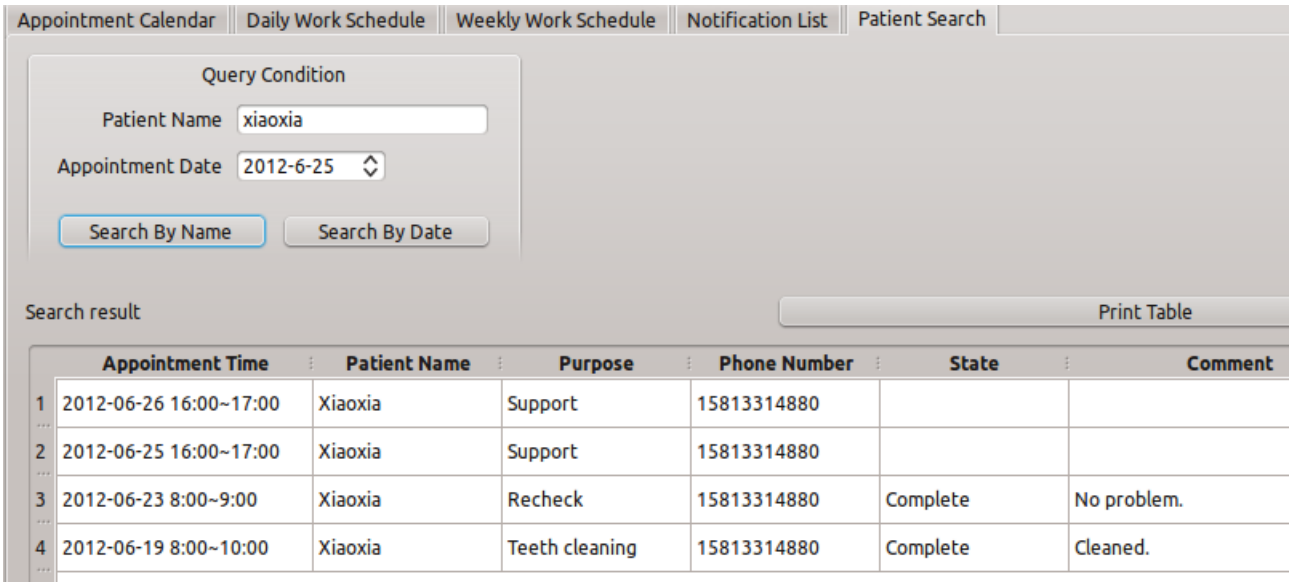
	Appointment Time	Patient Name	Purpose	Phone Number
1	2012-06-26 8:00~9:00	Chen Xiaodong	Teeth cleaning	15813312343
2	2012-06-26 14:00~15:00	Luo Yufeng	Tongue cleaning	15813312322
3	2012-06-26 15:00~16:00	Lian Huiqiong	Support	15813316544
4	2012-06-26 16:00~17:00	Xiaoxia	Support	15813314880
5	2012-06-27 8:00~9:00	Liang Kaidong	Teeth cleaning	15813316766
6	2012-06-27 14:00~15:00	Chen Xiaodong	Recheck	15813311223
7	2012-06-27 16:00~17:00	Zengshan	Teeth cleaning	15813316546

The *WeeklyWorkScheduleView* shows the work of the week.



	Work Time	Patient Name	Purpose	Phone Number	State	Comment
1	2012-06-23 8:00~9:00	Xiaoxia	Recheck	15813314880	Complete	No problem.
2	2012-06-23 9:00~10:00	Gu Enlai	Gum care	15813315234	Complete	Cleaned.
3	2012-06-23 10:00~11:00	Liu Zhiteng	Oral irrigation	15813312342	Complete	Checked no problem.
4	2012-06-23 16:00~17:00	Peng Feizhi	Teeth cleaning	15813312343	Complete	Cleaned.
5	2012-06-24 8:00~9:00	Lei Yu	Recheck	15813312141	Complete	No problem!
6	2012-06-24 14:00~16:00	Xie xinrong	Tongue cleaning	15813313243	Need Recheck	Tongue broken!!!
7	2012-06-25 8:00~10:00	Patrick Chan	Tongue cleaning	15813316543	Complete	Good tongue.

The *PatientSearchView* helps the hygienist to find patient records.



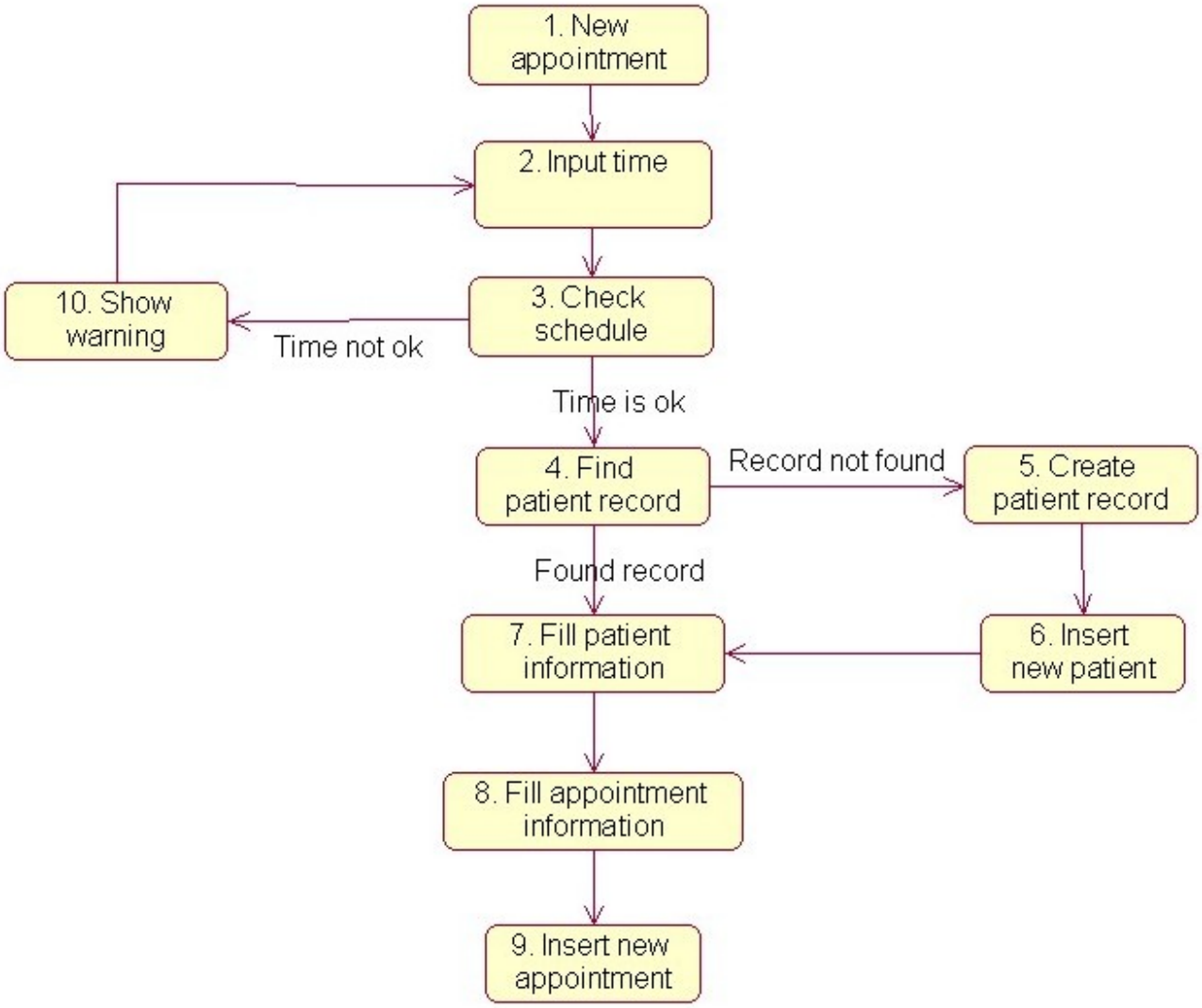
	Appointment Time	Patient Name	Purpose	Phone Number	State	Comment
1	2012-06-26 16:00~17:00	Xiaoxia	Support	15813314880		
2	2012-06-25 16:00~17:00	Xiaoxia	Support	15813314880		
3	2012-06-23 8:00~9:00	Xiaoxia	Recheck	15813314880	Complete	No problem.
4	2012-06-19 8:00~10:00	Xiaoxia	Teeth cleaning	15813314880	Complete	Cleaned.

Part 4. Test

Exercise 4.1

Draw a graph to show the structure (execution) of the ‘appointment’ program.

Solution



Exercise 4.2

Test case design for the ‘appointment’ program.

1. Should cover all statements (or methods) of the program.
2. Every branch should been exercised for true and false conditions.
3. List input and corresponding execution path.

Solution

The structural testing is used in this program. Path testing requires that each independent path through the program is executed at least once.

Assume that the appointment time is available except Tuesday 8:00 A.M. The patient name Xiaoxia is in the database while Patrick is not.

The execution path is as below.

Input	Execution Path
Time: Tuesday 8:00 A.M. **Patient:** Xiaoxia	1 → 2 → 3 → 10 → 2 → 3 → 4 → 7 → 8 → 9
Time: Monday 8:00 A.M. **Patient:** Patrick	1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 9

In this test case, two independent paths cover all the statements of the 'appointment' program. Therefore, the program is well tested.