

Noah Streveler

```
Are you a customer(enter 1) or staffer(enter 2)?
1
>Enter your first name
Kyle
>Enter your last initial
T
>Do you have an appointment(enter 1 for yes, 0 for no)
0
>Why are you here? Select one by entering the number:
    1. TV service
    2. Internet service
    3. Cell phone service
    4. Steam service
    5. Other
5
>You are set!
Are you a customer(enter 1) or staffer(enter 2)?
1
>Enter your first name
Noah
>Enter your last initial
S
>Do you have an appointment(enter 1 for yes, 0 for no)
1
>Why are you here? Select one by entering the number:
    1. TV service
    2. Internet service
    3. Cell phone service
    4. Steam service
    5. Other
2
>You are set!
Are you a customer(enter 1) or staffer(enter 2)?
2
>The front customer from the appointment line is
Noah S needs Internet Service
>Info about Noah S is removed
```

```
import java.util.*;
import javax.swing.*;

class AddressAppController implements AppController{
    AppView view;
    IAddressBook book;
    IDataHandler handler;
    String input, viewName, str;
    int job;
    LinkedList<String> reserve = new LinkedList<String>();
    LinkedList<String> walkup = new LinkedList<String>();

    AddressAppController(){
        appInit();
    }

    @Override
    public void setView(String name) {
        this.viewName = name;
    }
}
```

```

        if(viewName.equalsIgnoreCase("console"))
            view = new ConsoleAppView(book);
        else if(viewName.equalsIgnoreCase("gui"))
            view = new GUIAppView(book);
        else view = null;
        /*
        if(view != null){
            view.display( "Current names \n" + ((AddressBook)
book).getKeyset().toString() );
        }
        else System.exit(1);
        */
    }

    @Override
    public void appInit() {
        book = new AddressBook();
        handler = new DataHandler(book);
        handler.getData("addresses.txt");
    }

    @Override
    public void run() {
        do{
            job = view.getInt("Are you a customer(enter 1) or staffer(enter
2)?");

            if(job == 1){
                String fname = view.getInput(">Enter your first name");
                String lname = view.getInput(">Enter your last initial");
                boolean app = view.getApp(">Do you have an
appointment(enter 1 for yes, 0 for no)");
                int reason = view.getReason(">Why are you here? Select one
by entering the number:\n\t1. TV service\n\t2. Internet service\n\t3. Cell phone
service\n\t4. Steam service\n\t5. Other");
                str = fname + " " + lname;
                if(app) {
                    reserve.enqueue(str);
                    reserve.enqueue(Integer.toString(reason));
                }
                else {
                    walkup.enqueue(str);
                    walkup.enqueue(Integer.toString(reason));
                }
                //book.add(fname, lname, app, reason);
                view.display(">You are set!");
                //view.display( ((AddressBook)
book).getKeyset().toString());
            }
            else if(job == 2){
                if(reserve.isEmpty() == false) {
                    view.display(">The front customer from the
appointment line is");

                    view.present(reserve);
                    walkup.dequeue();
                }
            }
        }
    }

```

```

        else if(walkup.isEmpty() == false) {
            view.display(">The front customer from the walk-in
line is");

            view.present(walkup);
            walkup.dequeue();
        }
        else
            view.display("There is no one in line right now,
check back later.");
        System.exit(0);
    }
}while(true);
}
}

```

```

class ConsoleAppView implements AppView {
    private Scanner sc;
    String input;
    IAddressBook book;
    int job, reason, app;

    ConsoleAppView(IAddressBook b){
        book = b;
        sc = new Scanner(System.in);
    }

    @Override
    public String getInput(String prompt) {
        System.out.println(prompt);
        input = sc.next();
        //input = JOptionPane.showInputDialog(prompt);
        return input;
    }

    @Override
    public int getInt(String prompt) {
        System.out.println(prompt);
        job = sc.nextInt();
        if(job != 1 && job != 2) {
            System.out.println("I'm sorry, but the option that you chose is
not available, please try again");
            System.exit(0);
        }
        //input = JOptionPane.showInputDialog(prompt);
        return job;
    }

    @Override
    public int getReason(String prompt) {
        System.out.println(prompt);
        reason = sc.nextInt();
        if(reason > 5 && reason < 1) {
            System.out.println("I'm sorry, but the option that you chose is
not available, please try again");
            System.exit(0);
        }
    }
}

```

```

    }
    //input = JOptionPane.showInputDialog(prompt);
    return reason;
}

@Override
public boolean getApp(String prompt) {
    System.out.println(prompt);
    app = sc.nextInt();

    if(app != 1 && app != 0) {
        System.out.println("I'm sorry, but the option that you chose is
not available, please try again");
        System.exit(0);
    }
    else if(app == 1)
        return true;
    //input = JOptionPane.showInputDialog(prompt);
    return false;
}

@Override
public String getResult() {
    String result = book.getAddress(input.trim());
    return result;
}

@Override
public void display(String msg) {
    //JOptionPane.showMessageDialog(null, msg );
    System.out.println(msg);
}

@Override
public void present(LinkedList<String> linkedQueue) {
    //JOptionPane.showMessageDialog(null, msg );
    System.out.println(linkedQueue);
}
}

```

```

interface AppView {
    String getInput(String prompt);
    String getResult();
    void display(String msg);
    int getInt(String prompt);
    boolean getApp(String prompt);
    int getReason(String prompt);
    void present(LinkedList<String> linkedQueue);
}

```

```

interface AppController{

```

```

        void appInit();
        void setView(String viewName);
        void run();
    }

    interface IAddressBook {
        void add(String name, String addresses);
        int getSize();
        void remove(String name);
        boolean contains(String name);
        String getAddress(String name);
    }

    interface IDataHandler{
        void getData(String connectionStr);
        void saveData(String connectionStr);
    }

import java.util.ArrayList;

public class LinkedQueue<T> {
    private Node<T> front, rear;
    private int size;

    public LinkedQueue() {
        front = rear = null;
        size = 0;
    }

    public void enqueue(T item) {
        if(item == null) return;
        Node<T> newNode = new Node<T>(item);
        if(size == 0) {
            rear = front = newNode;
        }
        else {
            rear.setLink(newNode);
        }
        rear = newNode;
        size++;
    }

    public T dequeue() {
        T item = front.getInfo();
        front = front.getLink();
        size--;
        if(size == 0) {
            front = rear = null;
        }
    }
}

```

```

    }
    return item;
}

public boolean isEmpty() {
    return size == 0;
}

public void removeSecond() {
    if(size == 0) return;
    T temp = front.getInfo();
    front = front.getLink();
    size--;
    front.setInfo(temp);
}

public String process(int number) {
    String str = "";
    if(number == 1) {
        str = "needs TV Service";
    }
    else if(number == 2) {
        str = "needs Internet Service";
    }
    else if(number == 3) {
        str = "needs Cell Phone Service";
    }
    else if(number == 4) {
        str = "needs Steam Service";
    }
    else if(number == 5) {
        str = "needs some unknown Service";
    }

    return str;
}

public String toString() {
    String str = "";
    String temp = "", temp2 = "";
    Node<T> cursor = front;
    int count = 0;
    int service;
    while(count < 2) {
        if((count % 2) == 1) {
            temp2 = cursor.getInfo().toString();
            service = Integer.valueOf(temp2);
            str += process(service) + "\n";
        }
        else{
            temp = cursor.getInfo().toString() + " ";
            str += temp;
        }

        cursor = cursor.getLink();
    }
}

```

```

        count++;
    }
    str += ">Info about " + temp + " is removed";
    return str;
}
}

```

```

class Node<T>{
    private T info;
    private Node<T> link;

    public Node(T inf){
        this.info = inf;
        link = null;
    }

    public T getInfo() {
        return info;
    }

    public void setInfo(T info) {
        this.info = info;
    }

    public Node<T> getLink() {
        return link;
    }

    public void setLink(Node<T> link) {
        this.link = link;
    }
}

```