Assignment 2: Software Implementation

Alia Lootah

College of Interdisciplinary Studies, Zayed University

ICS220 - 22873 Programming Fundamentals

Professor Kuhail

April 15, 2023

**UML Class Diagram and Description:**

**Classes and Attributes:**

1. DentalBranch
   * address (private)
   * phoneNumber (private)
   * manager (private)
2. Staff
   * name (private)
   * id (private)
   * title (private)
3. Manager (inherits from Staff)
4. Receptionist (inherits from Staff)
5. Hygienist (inherits from Staff)
6. Dentist (inherits from Staff)
7. DentalService
   * serviceName (private)
   * serviceCost (private)
8. Patient
   * name (private)
   * id (private)
   * phoneNumber (private)
9. Appointment
   * appointmentID (private)
   * date (private)
   * time (private)
10. Bill
    * billID (private)
    * totalCost (private)
    * VAT (private)

**UML Class Diagram:**

|  |
| --- |
| DentalService |
| -serviceName: str  -serviceCost: float |

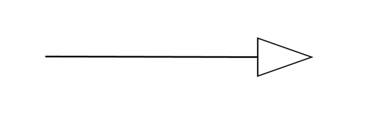
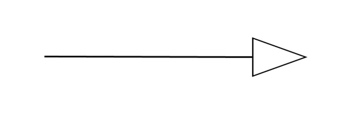
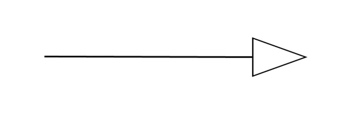
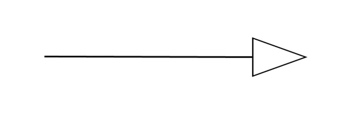
|  |
| --- |
| DentalBranch |
| -address: str  -phoneNumber: str  -manager: Manager |
| +add\_staff(staff: Staff)  +add\_patient(patient: Patient)  +add\_service(service: DentalService) |

|  |
| --- |
| Patient |
| -name: str  -id: str  -phoneNumber:str |
| +book\_appointment(appointment: Appointment) |



|  |
| --- |
| Manager |

|  |
| --- |
| Staff |
| -name: str  -id: str  -title: str |



|  |
| --- |
| Hygienist |

|  |
| --- |
| Dentist |

|  |
| --- |
| Receptionist |

|  |
| --- |
| Patient |
| -name: str  -id: str  -phoneNumber:str |
| +book\_appointment(appointment: Appointment) |

|  |
| --- |
| Appointment |
| -appointmentID:str  -date: Date  -time: Time  -dentist: Dentist  -services: List[DentalService] |
| +generate\_bill(): Bill |

|  |
| --- |
| Bill |
| -billID: str  -totalCost: float  -VAT: float |
| +display\_receipt() |

**Relationships:**

1. DentalBranch has a 1-to-many relationship with Staff, as one branch can have multiple staff members.
2. DentalBranch has a 1-to-many relationship with DentalService, as one branch can offer multiple services.
3. DentalBranch has a 1-to-many relationship with Patient, as one branch can have multiple patients.
4. Staff has a generalization relationship with Manager, Receptionist, Hygienist, and Dentist. These classes inherit from the Staff class.
5. Patient has a 1-to-many relationship with Appointment, as one patient can have multiple appointments.
6. Appointment has a 1-to-1 relationship with Dentist, as one appointment is linked to a single dentist.
7. Appointment has a 1-to-many relationship with DentalService, as multiple services can be provided during one appointment.
8. Bill has a 1-to-1 relationship with Appointment, as one bill is linked to one appointment.

The diagram also shows the relationships between these classes, including inheritance, composition, and association relationship

1. Inheritance: Manager, Receptionist, Hygienist, and Dentist classes inherit from the Staff class. This relationship is shown by the arrows pointing from the derived classes to the base class.
2. Composition: DentalBranch class has a composition relationship with the Staff class, as it has an attribute of type Manager, which is a subclass of Staff. This is represented by the diamond symbol next to the DentalBranch class and the line connecting it to the Staff class.
3. Association:
   * DentalBranch has associations with Patient and DentalService classes, indicated by the add\_patient, add\_service, and add\_staff methods. This relationship shows that a DentalBranch keeps track of its patients and services.
   * Appointment has associations with Dentist and DentalService classes. The dentist attribute and services list in the Appointment class represent these relationships. This indicates that an appointment has a dentist assigned to it and involves one or more dental services.
   * Patient has an association with Appointment, represented by the book\_appointment method. This relationship shows that a patient books an appointment at the dental clinic.
   * Appointment has an association with Bill, represented by the generate\_bill method. This relationship indicates that a bill is generated for an appointment after the services are completed.

**Assumptions:**

1. Staff can be uniquely identified by their 'id' attribute.
2. Patients can be uniquely identified by their 'id' attribute.
3. Each dental service has a unique 'serviceName' and a corresponding 'serviceCost'.
4. Appointments can be uniquely identified by the 'appointmentID' attribute.
5. Bills can be uniquely identified by the 'billID' attribute.
6. The 5% VAT is a constant value in the Bill class.