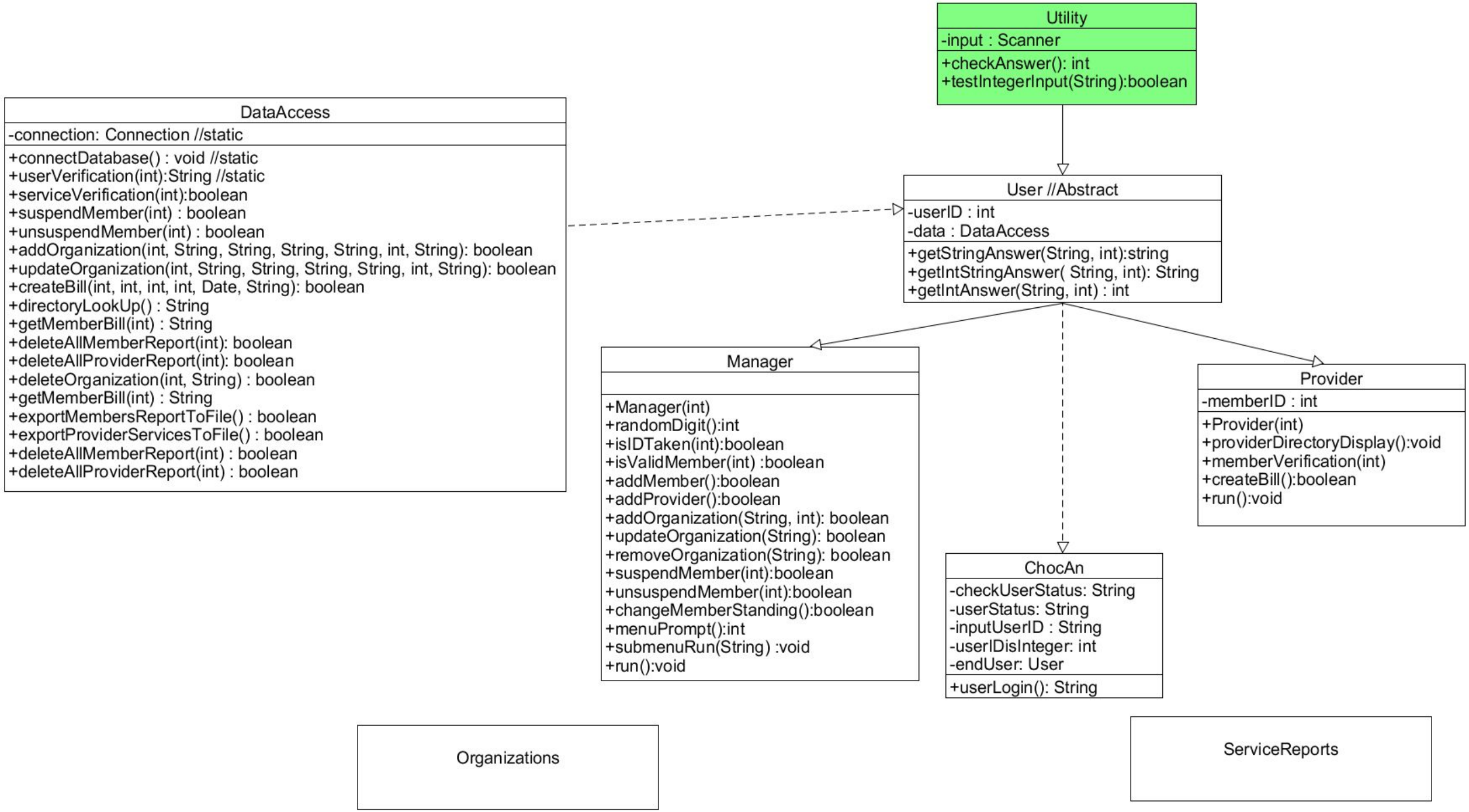
Chocoholics Anonymous

Gr8 Software Solutions


Utility
-input : Scanner
+checkAnswer(): int
+testIntegerInput(String):boolean
DataAccess
-connection: Connection //static
+connectDatabase() : void //static
+userVerification(int):String //static
+serviceVerification(int):boolean
+suspendMember(int) : boolean
+unsuspendMember(int) : boolean
+addOrganization(int, String, String, String, String, int, String): boolean
+updateOrganization(int, String, String, String, String, int, String): boolean
+createBill(int, int, int, int, Date, String): boolean
+directoryLookUp() : String
+getMemberBill(int) : String
+deleteAllMemberReport(int): boolean
+deleteAllProviderReport(int): boolean
+deleteOrganization(int, String) : boolean
+getMemberBill(int) : String
+exportMembersReportToFile() : boolean
+exportProviderServicesToFile() : boolean
+deleteAllMemberReport(int) : boolean
+deleteAllProviderReport(int) : boolean
User //Abstract
-userID : int
-data : DataAccess
+getStringAnswer(String, int):string
+getIntStringAnswer(String, int): String
+getIntAnswer(String, int) : int
Manager
+Manager(int)
+randomDigit():int
+isIDTaken(int):boolean
+isValidMember(int) :boolean
+addMember():boolean
+addProvider():boolean
+addOrganization(String, int): boolean
+updateOrganization(String): boolean
+removeOrganization(String): boolean
+suspendMember(int):boolean
+unsuspendMember(int):boolean
+changeMemberStanding():boolean
+menuPrompt():int
+submenuRun(String) :void
+run():void
Provider
-memberID : int
+Provider(int)
+providerDirectoryDisplay():void
+memberVerification(int)
+createBill():boolean
+run():void
ChocAn
-checkUserStatus: String
-userStatus: String
-inputUserID : String
-userIDisInteger: int
-endUser: User
+userLogin(): String
Organizations
ServiceReports

Functionality

- Main
  - Contains a ChocAn object that, upon starting the program simply calls its run() method.
  - Keeps code clean and keeps our contribution to the system modular.
- ChocAn
  - Designed to make the calls to the appropriate methods that follow the flow of execution described in the Specification document and the contract awarded to Group 8.
  - Used to assign the appropriate role of the user based on the results returned during login.
  - Creates a layer of encapsulation and abstraction; provides better options for code evolution.
- Utility class
  - Provided a ‘tool box’ for user input testing and prompts throughout development.
  - Allowed us to extend to classes with automatic access to a Scanner and Utility methods.

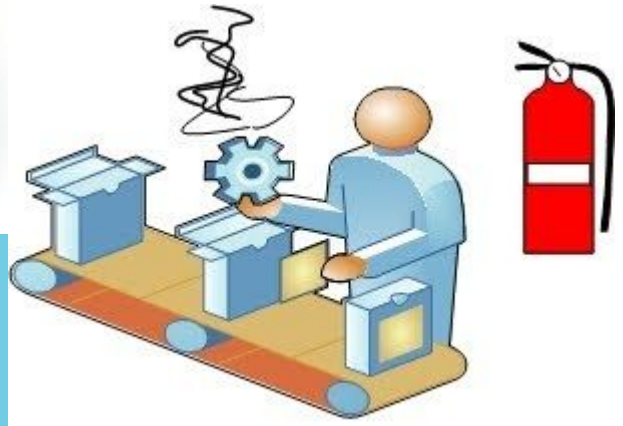
Functionality

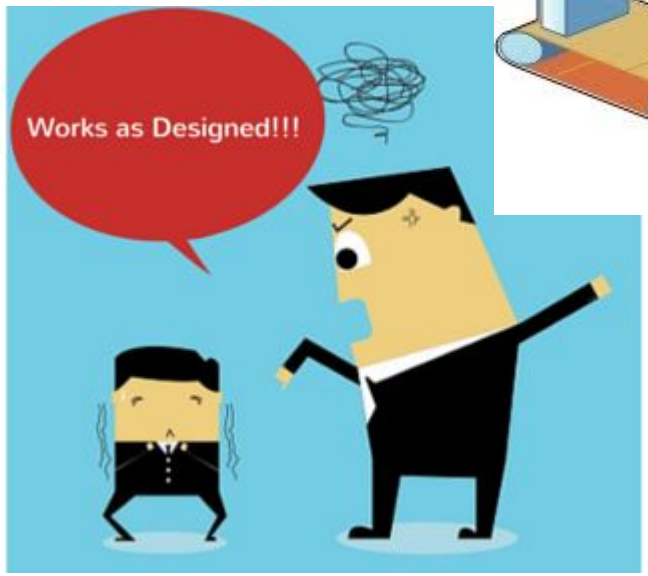
- DataAccess
  - Acts as the main form of communication between the database and system users.
  - Implements methods to create, edit, and remove information from the database.

- User class
  - Abstract User class allowed us to have one user object in ChocAn, allocated appropriately based on the result of DataAccess.userVerification() static method.
  - Abstract run() method allowed us to define the flow of the menu functionality in Manager and Provider class.
  - Contains a DataAccess object that allows communication between the end-user and the database.
- Manager/Provider class
  - Uses abstract run() method as a menu, presenting options corresponding to the required tasks of a ChocAn manager/provider dependant upon user type.
  - Implements methods to call its DataAccess object's methods to perform these tasks.

Testing

- Unit Test
  - Junit4
  - Are methods returning expected output?
- Smoke Test
  - Manually
  - Is functionality working properly for user inputs?
- System Test
  - Manually
  - How subsystems come together as a final product?





Functionality issues

- Implementing a menu to prompt for the type of information a manager would like to change regarding a provider or member. This would allow the system user to change specific information regarding a provider or member instead of all information.
- Lacks the ability to retrieve a week worth of provider and member reports. Currently fetches all provider and member reports.
- To account for the “emailing” functionality we just export the records to an external text file which can then be viewed.
- Currently only generates member reports for members who have active status.