Team 12

Cs200 Spring 2022

Dr. Smith, Bryan Hernandez

4/29/22

Project 4 Short User Manual

Our software, consisting of 18 classes, has been implemented in a way to demonstrate high cohesion and low coupling, as our classes work together to build the required ChocAn software. We start with a MainMenu class, which is the basic driver code to allow the user to use the software. It allows the user to choose their “user type”, whether that be a Provider, Manager or Operator. Once this decision is made, the user is directed to the methods in MainMenu that are specific to the user type, namely ProviderLogin, ManagerLogin and OperatorLogin. These methods prompt the user for their credentials (username/email and password/ID). Then the class for the user-type is called along with the user-type list in order to verify the user’s credentials using methods inside the user/user-list classes such as isProvider(), isManagerPass(), etc.

Once the user is verified through the login methods in MainMenu, the menu class of the specified user type is called. We have menu classes for each user, a ProviderMenu class, ManagerMenu class and OperatorMenu class. Each of these three contain the interfaces that allow the specified user to interact with the system and the methods that are allowed for their user type.

If the user logs in as a provider, they are sent to the ProviderMenu. The ProviderMenu class then displays all of the actions the provider can perform: Validate Member, display the Provider Directory, or Bill ChocAn for their provided services. Once this decision is made, the class calls the specified methods and allows the user to perform whichever action they choose.

If the user logs in as a manager, they are directed to the ManagerMenu where all the actions a manager can perform in the system are displayed. The manager can choose to create a singular report (Provider, Member, EFT, or summary), or create all four reports at the same time for the end-of-week ChocAn procedure. The ManagerMenu class then calls the Manager class and calls the method to create the specified reports requested by the manager.

If the user logs on as an operator, the OperatorMenu is called and the actions that can be performed by an operator are displayed. The operator is the least involved, as their primary duties are adding, deleting or editing either a member or a provider. The operator is prompted to manage either a member or provider, and then the OperatorMenu calls the methods needed based on the operator’s choice. The operator then chooses to add, delete or update the member/provider. The OperatorMenu calls the Provider/Member classes and methods to be able to access their data and lists, to be able to manage the provider or members effectively.

At any time, the user that is interacting with the system can exit the system by entering the specified exit input, which varies from class to class, but is always available. Once the user has completed their task and is done with the system, they can enter the appropriate logout message and return to the MainMenu, where they can exit the system.

|  |  |  |  |
| --- | --- | --- | --- |
| Group Member Name | Tasks/Classes Worked On | Hours Contributed | % Contributed |
| Quillen Flanigan | Provider, Provider List, Provider Menu, Short Manual, Task Distribution table, ANT configuration, JAR configuration | 18 | 16% |
| Liam Tucker | Provider Menu, Service, Service List, Service Name Database, Member List, Extensive Testing, Javadoc | 25 | 22% |
| Deondre North | Main, Main Menu, GUI | 20 | 18% |
| Brandon Jones | Manager, Manager List, Manager Menu, Extensive Testing, Main Menu | 22 | 20% |
| Ramsey Morrow | Operator, Operator List, Operator Menu | 15 | 13% |
| Keaton Rohlfs | Member, Member List | 15 | 13% |