## **Business Enterprise Rules**

[Abstract or Interface or Normal]	vent	[Enter Super Class Here]
_	Vent	[Enter Subclass Here]
Responsibilities	• Collaborators	
<ul> <li>Stores the name, description, and times of an event.</li> <li>Has methods to edit the events and the event times.</li> </ul>	• Schedule	

Normal Sch	nedule	[Enter Super Class Here]
	Slee	epSchedule, MedicineSchedul
Responsibilities	• Collaborators	
<ul> <li>Has a list of events.</li> <li>Be add and remove events.</li> <li>Tracks both day and hour for the time of an event.</li> </ul>	TimeTracker     Event	

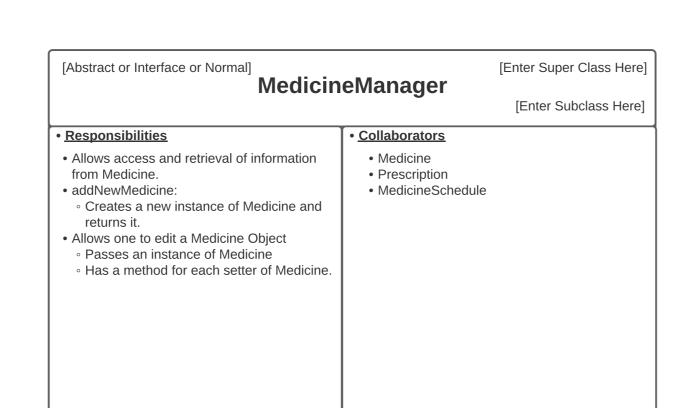
Normal <b>Me</b> (	dicine	[Enter Super Class Here]
Decree de illaire	O allah a matawa	PrescriptionMedicine
<ul><li><u>Responsibilities</u></li><li>Store the name, expiration date, amount,</li></ul>	• <u>Collaborators</u> • Prescription	
<ul> <li>type, method of administration and extra instructions.</li> <li>Stores an integer ID for the medicine.</li> <li>Stores the alternate names of the medicine.</li> <li>Tracks whether the medicine is expired or</li> </ul>	MedicineSch	edule
<ul> <li>Stores which medicine this specific medicine was prescribed with.</li> <li>Stores a Medicine schedule for this specific medicine.</li> <li>Getters and Setters for all the above variables.</li> </ul>		

Normal	eSchedule	Schedule
Medicili	Cochedule	[Enter Subclass Here]
Responsibilities	• Collaborators	
<ul> <li>Stores the name of a medicine and ID of the medicine.</li> <li>Stores the number of times to take a medicine per day</li> <li>Stores the times to take a medicine.</li> <li>Done by Event</li> </ul>	Medicine	

[Abstract or Interface or Normal]		[Enter Super Class Here]
	[Enter Subclass Here]	
Responsibilities	• <u>Collaborators</u>	
<ul> <li>Stores username and name of the user as private attributes.</li> <li>Stores a list of all the medicine they have.</li> <li>Stores instances of Medicine</li> <li>Getters and Setters for these values</li> <li>Capable of adding and removing instances of medicine.</li> </ul>	UserManager     Medicine	

## **Application Business Rules**

[Abstract or Interface or Normal]  UserManager		[Super Class]
		[Enter Subclass Here
• Responsibilities	• Collaborators	
<ul> <li>Allows one to edit information in the User class</li> <li>Stores an instance of User</li> <li>Allows one to manage the medicine for a specific user.</li> <li>Stores an instance of MedicineManager.</li> <li>Calls user to find specific medicine to pass into MedicineManager to edit.</li> <li>addNewUser:</li> <li>Creates a new User instance with username, and name.</li> <li>Stores this User instance as an attribute.</li> <li>Returns the User instance.</li> <li>Allows one to add medication by calling MedicineManager.</li> </ul>	User     ManagementSystem	

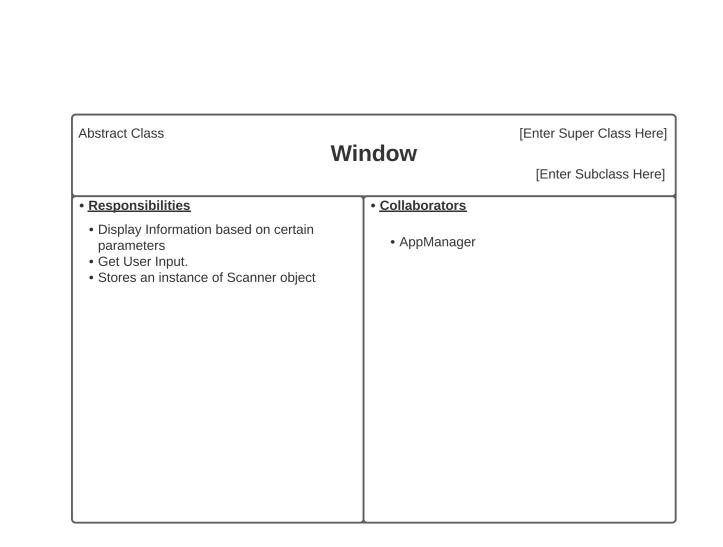


Normal	Schedu	leManager	[Enter Super Class Her
ScheduleManager		[Enter Subclass Her	
• Responsibilities		• Collaborators	
Stores the schedule may the other schedules. Ear own MedicineSchedule of these schedules, we combined one in Schedule editSchedule. Allows or Schedule objects and it compileSchedule: Uses to compile a bunch of Spassed in as parameter. Returns the schedule.	ach medicine has it's . After compliling all store the final, luleManager. ne to edit individual 's Events. s ScheduleCompiler schedule instances	• Schedule	

Abstract or Interface or Normal] ScheduleCompiler		[Enter Super Class He
ScheduleCompilei		[Enter Subclass Her
• Responsibilities	• Collaborators	
<ul> <li>Creates a Schedule using the other schedules that were passed in as parameters. Returns this schedule.</li> </ul>	• Schedule	

Normal		[Enter Super Class Here]
Manage	mentSystem	
		[Enter Subclass Here]
<u>Responsibilities</u>	• Collaborators	
<ul> <li>Stores a map of usernames to User instances.</li> <li>Manages interactions between interface adapters and application business rules.</li> <li>Stores instances of UserManager, ScheduleManager, AlertSystem.</li> <li>Takes in username from AppManager and finds the user entity for that specific user using the map. Set that User as the User in UserManager.</li> <li>addNewUser: <ul> <li>Calls UserManager.addNewUser. Use the returned instance and add that to the map with the username.</li> <li>getUserInfo:</li> <li>Uses UserManager to return the user's name, username, and the list of medicine (names of the medicines) the user has.</li> <li>getFinalSchedule:</li> <li>Calls UserManager to get a list of Medicine.</li> <li>Calls ScheduleManager to get a final schedule and passes in the list of MedicineSchedule it got from the list of Medicine.</li> </ul> </li> <li>Acts as an intermediary between UserManager, ScheduleManager and AppManager. Thus is responsible for retrieving and sending information between these two entities.</li> </ul>	UserManager ScheduleManager AlertSystem AppManager	

## **Interface Adapters**



Normal		InputOutputManager
AppManager		
		[Enter Subclass Here]
Responsibilities	• Collaborators	
Keeps track of all the usernames and passwords		
<ul> <li>Able to add new usernames and passwords, as well as remove old usernames and passwords.</li> </ul>		
If the account exists, the user is passed to ManagementSystem	ManagementSystem	
Creates an instance of:	,	
StartScreenWindow     A print Window		
<ul><li>LoginWindow</li><li>CreateAccountWindow</li></ul>		
• TimeTable		
Creates an instance of ManagementSystem. Passes this instance as a parameter to		
ManagementSystem.		
Has a run method that runs the program.		
createNewAccount:		
Calls CreateAccountWindow using the Window Interface and displays options		
• gets user input on their name, username and password.		
<ul> <li>CreateAccountWindow returns what the user entered for their name, username and password.</li> </ul>		
• showAccountWindow:		
• Calls ManagementSystem to get the user's name, username, and list of medicine they have.		
• Call ViewAccountWindow using the Window Interface to show the user information.		
<ul> <li>ViewAccountWindow returns user input.</li> </ul>		
$_{\circ}$ Based on user input, we can edit the medicines and medicine list, edit user information, view		
the time table and logout (not all are implemented in phase 0)		
• editMedicine:		
Calls the EditMedicineWindow using the Window Interface     Calls ManagementSystem to edit the medicine		
<ul> <li>Calls ManagementSystem to edit the medicine.</li> <li>removeMedicine:</li> </ul>		
Calls the RemoveMedicineWindow using the Window Interface		
• Calls ManagementSystem to remove the new medicine.		
addMedicine:		
<ul> <li>Calls the AddMedicineWindow using the Window Interface</li> </ul>		
<ul> <li>Calls ManagementSystem to add the new medicine for the user.</li> </ul>		
• getFinalSchedule:		
Calls ManagementSystem to get a Schedule.		
<ul> <li>logOut:</li> <li>Allows users to logout and go back to login page.</li> </ul>		
, mone ascrs to logout and go back to login page.		

## Frameworks & Drivers

cinewindow	[Enter Subclass Here]	CreateAC
• Collaborators		Responsibilities
• [Enter List Here]		Shows the options of entering a name, username and password.     Returns the user input of their name, username and password.
icineWindow	Window [Enter Subclass Here]	Normal <b>RemoveM</b>
• Collaborators		Responsibilities
		Allows users to remove old medicine
	• [Enter List Here]  icineWindow  • Collaborators	• Collaborators • [Enter List Here]  Window  [Enter Subclass Here]  • Collaborators • [Enter Subclass Here]  • Collaborators • [Enter List Here]

EditMedicineWindow

Normal

Normal <b>RemoveMe</b>	edicineWindow	Window
Kemovelvie	, archite vvii i aovv	[Enter Subclass Here]
Responsibilities	• Collaborators	
Allows users to remove old medicine	• [Enter List Here]	

CreateAccountWindow

Collaborators

• [Enter List Here]

[Enter Subclass Here]

bstract or Interface or Normal]  LoginWindow		Window	
<b>9</b>	Logiiiviiidov		
Responsibilities	• Collaborators		
<ul><li>Allows the user to enter their username and password to sign in</li><li>Takes in user input</li></ul>	MenuManager		

Responsibilities	• Collaborators
showOptions:         The user has two options:         Login         Signup     getInput:         Gets an integer input for which option is picked. Return that input.	LoginWindow     MainMethod     CreateAccountWindow
Normal	[Enter Super Clas

Normal <b>TimeTableWindo</b>		Window
Timera	[Enter Subclass Here	
• Responsibilities	• Collaborators	
<ul> <li>Displays a visual representation of the schedule.</li> <li>Returns an integer if the user wants</li> </ul>	Schedule	
to go back to ViewAccountWindow using the user input.		

Normal	untWindow	Window [Enter Subclass Here]	Normal	[Enter Super Class H
• Responsibilities	• Collaborators		• Responsibilities	• Collaborators
<ul> <li>Shows the username, name, and a list of medication the user uses.</li> <li>Allows users to select whether they want to add new medicine, remove a medicine, edit their medicine or look at their medication timetable.</li> <li>Obtains user input.</li> <li>Allows the user to log out.</li> </ul>	• [Enter List Here]		Has the main method to run the program.     Creates an instance of AppManager and instances of all the classes that implemer Window and passes them into AppManager's run method.	Window and it's implementations     AppManager  It