**CLASS: ScheduleList**

**Summary:**

**Elements:**

1. private List<Schedule> Schedules

**Functions:**

1. public boolean AddSchedule(schedule newSchedule)
2. public Schedule GetSchedule(integer reportType)
3. public Schedule UpdateSchedule(schedule updatedSchedule)
4. public boolean DeleteSchedule(integer reportType)

**Detail:**

1. **public boolean AddSchedule(schedule newSchedule)**

Module name : **AddSchedule**

Module type : Function

Return type : **boolean**

Input arguments : **newSchedule : schedule**

Output arguments : None

Error messages : file access error, invalid schedule

Files accessed : schedules.dat

Files changed : schedules.dat

Modules called : None

Narrative : This method adds a new schedule information.

public boolean AddSchedule(schedule newSchedule)

{

boolean success = false;

try

{

if (!ValidateSchedule(newSchedule))

throw new Exception(“invalid schedule”);

if (Schedules

.Select(s => s.GetReportType() == newSchedule.GetReportType())

.Any())

throw new Exception(“invalid schedule”);

//add new schedule to schedules.dat

//refresh Schedules

Success = true;

}

catch (Exception e)

{

//record exception

Success = false;

}

return success;

}

1. **public Schedule GetSchedule(integer reportType)**

Module name : **GetSchedule**

Module type : Function

Return type : **Schedule**

Input arguments : **reportType : integer**

Output arguments : None

Error messages : file access error

Files accessed : schedules.dat

Files changed : None

Modules called : None

Narrative : This method returns the target report schedule

public Schedule GetSchedule(integer reportType)

{

Schedule schedule = new Schedule();

try

{

schedule = Schedules

.Select(s => s.GetReportType() == reportType).FirstOrDefault();

}

catch (Exception e)

{

//record exception

Schedule = null;

}

return Schedule;

}

1. **public Schedule UpdateSchedule(schedule updatedSchedule)**

Module name : **UpdateSchedule**

Module type : Function

Return type : **Schedule**

Input arguments : **updatedSchedule : schedule**

Output arguments : None

Error messages : invalid schedule

Files accessed : schedules.dat

Files changed : schedules.dat

Modules called : None

Narrative : This method updates current schedule information.

public Schedule UpdateSchedule(schedule updatedSchedule)

{

Schedule newSchedule = new Schedule();

try

{

if (!ValidateSchedule(updatedSchedule))

throw new Exception(“invalid schedule”);

if (!Schedules

.Select(s => s.GetReportType() == newSchedule.GetReportType())

.Any())

throw new Exception(“invalid schedule”);

//update schedule in schedules.dat

//refresh Schedules

newSchedule = //get updated schedule;

}

catch (Exception e)

{

//record exception

newSchedule = null;

}

return newSchedule;

}

1. **public boolean DeleteSchedule(integer reportType)**

Module name : **DeleteSchedule**

Module type : Function

Return type : **boolean**

Input arguments : **reportType : integer**

Output arguments : None

Error messages : invalid schedule

Files accessed : schedules.dat

Files changed : schedules.dat

Modules called : None

Narrative : This method deletes a current schedule information.

public boolean DeleteSchedule(schedule newSchedule)

{

boolean success = false;

try

{

//delete schedule from schedules.dat

//refresh Schedules

Success = true;

}

catch (Exception e)

{

//record exception

Success = false;

}

return success;

}