Code Documentation

Michael Kramarczyk
CIS 631
mdk5@njit.edu

Class: schedule

The schedule class contains the main class in the program. It manages the desktop GUI environment and includes the handling for the mouse click response for the main menu bar:

- Main Menu: Login, Logoff, Exit
- Reports: Staff Report, Employee Report
- Admin: Schedule (Schedule and Vacation), Employee (Employee and Employee Job), and Job
- Help: Help, Test Database Connection, About

schedule()
This is the constructor which initializes the class settings.

void createWindow(final Component newContentPane, final String window, final int size1, final int size2,final boolean closeable, final boolean maxSize)
This is the main function for launching the windows into the desktop. The component newContentPane is the window pane content. The String window is the title for the window. The size1 is the X axis size. The size2 is the Y axis size. If boolean colesable is true then the window can be closed. If the boolean maxSize is true then the size can not be changed.

void createHelp()
The function used for creating the help window. The help class is called to create the actual window.

void createLogin()
The function used for creating the login window. The login class is called to create the actual window.

void createAddEmployee()
The function used for creating the employee window. The schedule_employee class is called to create the actual window.

void createAddJob()
The function used for creating the job window. The schedule_job class is called to create the actual window.

void createEmployeeJob()
The function used for creating the employee job window. The schedule_employeeJob class is called to create the actual window.

void createAddVacation()
The function used for creating the vacation window. The schedule_vacation class is called to create the actual window.

void createSchedule()
The function used for creating the schedule window. The schedule_schedule class is called to create the actual window.

void createPrintEmployee()
The function used for creating the employee report window. The schedule_PrintEmployee class is called to create the actual window.

void createPrintStaff()
The function used for creating the staff report window. The schedule_PrintStaff class is called to create the actual window.

void createDesktop()
The function used for creating the desktop. This function will call the constructor in the mainMenuBar class.

static void main(String[] args)
This is the main function in the program, which begins by calling the createDesktop function.

Subclass: mainMenuBar

The schedule class contains the mainMenuBar subclass which handles the main menu bar. This class contains functions for creating the main menu bar, handling the main menu bar input, and calling the functions to launch the coordinating windows.

mainMenuBar()
This is the constructor which initializes the class settings, and creates the menu bar.

void actionPerformed(ActionEvent e)
This function handles the input for the mainMenuBar.

Subclass: login

The schedule class contains the login subclass which handles the login screen. This class contains functions for creating the login GUI, handling the GUI input, and validating the username, password combination.

login()
This is the constructor which initializes the class settings and creates the login screen.

void actionPerformed(ActionEvent e)
This is the function, which handles the GUI input for the login screen.

char userCheck()
This is the function, which checks if the user name field is blank. If an error is detected, an error message is generated.

char passwordCheck()
This is the function, which checks if the password field is blank. If an error is detected, an error message is generated.

void userlogin()
This is the function, which calls schedule_database functions to check the validity of the username and password combination. If it is not valid, an error message is printed out. If it is valid, a confirmation message is printed out and the user is logged in and given the proper user rights.

Class: schedule_database

The schedule_database class contains the functions which handle all the Microsoft Access Database read and writes. All the SQL queries used in the program are contained in this class. Whenever a function needs to access the database, it must create a schedule_database object. There are many functions which include:

void schedule_database()
This is the constructor which sets up the initial conditions.

void createTables ()
This function creates the employee, employeeJob, Job, schedule, and vacation tables.

record_Employee insertEmployee (record_Employee R)
This function inserts an employee record into the employee table and returns the employee record. It will also report any errors, which occur.

record_Job insertJob (record_Job R)
This function inserts a job record into the job table and returns the job record. It will report any errors which occur.

record_EmployeeJob insertEmployeeJob (record_EmployeeJob R)
This function inserts an employee job record into the employee job table, and returns the employee job record. It will report any errors which occur.

record_Schedule insertEmployeeSchedule (record_Schedule R)
This function inserts an employee schedule record into the employee schedule table and returns the employee schedule record. It will report any errors which occur.

record_Vacation insertEmployeeVacation (record_Vacation R)
This function inserts an employee vacation record into the employee vacation table and returns the employee vacation record. It will report any errors which occur.

record_Employee updateEmployee (record_Employee R)

This function updates an employee record in the employee table and returns the employee record. It will report any errors which occur.

record_Job updateJob (record_Job R)
This function updates a job record in the job table and returns the job record. It will report any errors which occur.

record_EmployeeJob updateEmployeeJob (record_EmployeeJob R)
This function updates an employee job record in the employee job table and returns the employee job record. It will report any errors which occur.

record_Schedule updateEmployeeSchedule (record_Schedule R)
This function updates an employee schedule record in the employee schedule table and returns the employee schedule record. It will report any errors which occur.

record_Vacation updateEmployeeVacation (record_Vacation R)
This function updates an employee vacation record in the employee vacation table and returns the employee vacation record. It will report any errors which occur.

String[] ListAllEmployees()
This function creates a list of all the employees entered into the system with the format:
Last Name, First Name, Employee Number.

String[] ListAllActiveEmployees()
This function creates a list of all the active employees entered into the system with the format:
Last Name, First Name, Employee Number.

String [][] PrintListAllActiveEmployees()
This function creates a printed list of all the employees entered into the system with the following fields:
Last Name, First Name, Employee Number, Jobs that the employee can do, and phone number.

String [][] PrintListAllActiveEmployees(int job_id)
This function creates a printed list of all the active employees entered into the system with the following fields:
Last Name, First Name, Employee Number, Jobs that the employee can do, and phone number.

String [][] PrintListAllInActiveEmployees()
This function creates a printed list of all the inactive employees entered into the system with the following fields:
Last Name, First Name, Employee Number, Jobs that the employee can do, and phone number.

String [][] PrintListAllActiveEmployees(int job_id)
This function creates a printed list of all the active employees entered into the system who perform a certain type of job with the following fields:
Last Name, First Name, Employee Number, Jobs that the employee can do, and phone number.

String [][] ListAllActiveEmployeesAvailable(int dateS)
This function creates a printed list of all the active employees entered into the system who are not working on a particular date with the following fields:
Last Name, First Name, Employee Number, Jobs that the employee can do, and phone number.

String [][] ListAllActiveEmployeesAvailable(int dateS, int job_id)
This function creates a printed list of all the active employees entered into the system who are not working on a particular date and who can perform a certain type of job with the following fields:
Last Name, First Name, Employee Number, Jobs that the employee can do, and phone number.

String [] ListAllActiveEmployeesWorking (int dateS)
This function creates a printed list of all the active employees entered into the system who are working on a particular date with the following fields:
Last Name, First Name, Employee Number, Jobs that the employee can do, and phone number.

String [] ListAllActiveEmployeesWorkingOptions (int dateS, int CMD)
This function creates a printed list of all the active employees entered into the system who are working on a particular date with the following fields:
Last Name, First Name, Employee Number, Jobs that the employee can do, and phone number.
The CMD contains a list of options such as:
ORDER by start time, job name employee name
ORDER by start time, employee name, job name
ORDER by job name, start time, employee name
ORDER by job name, employee name, start time
ORDER by employee name, start time, job name
ORDER by employee name, job name, start time

public String [] ListAllActiveEmployeesWorkingOptions (int dateS, int job_id, int CMD)
This function creates a printed list of all the active employees entered into the system who are working on a particular date who can work a particular job with the following fields:
Last Name, First Name, Employee Number, Jobs that the employee can do, and phone number.
The CMD contains a list of options such as:
ORDER by start time, job name employee name

ORDER by start time, employee name, job name
ORDER by job name, start time, employee name
ORDER by job name, employee name, start time
ORDER by employee name, start time, job name
ORDER by employee name, job name, start time

String [] ListAllActiveEmployeesWorking (int dateS, int job_id)
This function creates a printed list of all the active employees entered into the system who are working on a particular date doing a particular job with the following fields: Last Name, First Name, Employee Number, Jobs that the employee can do, and phone number.

String[] ListAllInActiveEmployees()
This function creates a printed list of all the inactive employees entered into the system with the following fields:
Last Name, First Name, Employee Id

String[] ListAllJobs()
This function creates a list of all the jobs in the system.

String [] ListAllActiveJobs()
This function creates a list of all the active jobs in the system

String[] ListAllEmployeeJobsWithEmployeeId(record_Employee RE)
This function creates a list of all the active employee jobs that an employee can work.

String[] ListAllEmployeeJobsWithEmployeeIdActive(record_Employee RE)
This function creates a list of all the active employee jobs that an active employee can work.

String[] ListAllEmployeeJobsWithEmployeeIdInActive(record_Employee RE)
This function creates a list of all the active employee jobs that an inactive employee can work.

record_Job getJobName(record_Job J)
This function returns the job record if supplied with the job id.

record_Job getJobId(record_Job J)
This function returns the job record if supplied with the job name.

record_Employee getEmployeeByEmployeeId(record_Employee R)
This function returns the employee record if given the employee id.

record_Employee getEmployeeWithUserName(record_Employee R)
This function returns the employee record if given the user name.

record_EmployeeJob getEmployeeJobWithEmployeeIdJobId(record_EmployeeJob R)
This function returns the employee job record if it exists when supplied with a possible employee job record.

record_Schedule getScheduleWithPositionDateS(record_Schedule R)
This function returns the schedule record if supplied with the position and dateS attributes.

record_Vacation getVacationWithPositionDateS(record_Vacation R)
This function returns the vacation record if supplied with the position and dateS attributes.

record_Schedule okSchedule(record_Schedule R)
This function checks if a person can be scheduled to work on a particular day.

record_Vacation okVacation(record_Vacation R)
This function checks if a person can be scheduled for vacation on a particular day.

record_Schedule getEmployeeScheduleWithEmployeeIdDateSActive(record_Schedule R)
This function returns an active employee schedule record if supplied with an employee_id, and DateS.

record_Vacation getEmployeeVacationWithEmployeeIdDateSActive(record_Vacation R)
This function returns an active employee vacation record if supplied with an employee_id, and DateS.

record_Employee getEmployeeWithUserNamePassword(record_Employee R)
This function checks if the username and password is legal combination. If it is then return the employee record.

noAdmin()
This function checks if there is an administrator in the system.

Class: schedule_employee

The schedule_employee class contains the functions for managing the Employee GUI, Employee GUI input, and for calling the database functions to insert/update employee records.
The following employee record attributes can be manipulated from this interface: Active, Admin Rights, User Name, Password, First Name, Last Name, Phone Number, and Cell Phone Number.
The User Name can not be modified once it is entered into the system. (Note: employee_id is hidden.)

schedule_Employee()

This is the constructor for the schedule_employee class. It begins by calling the findEmployee subclass.

Subclass: findEmployee

This class is responsible for creating the initial screen for the employee add/edit window. Here the user can select a user from a list of employees in the system, and select the options: Add Employee, Edit Employee.

findEmployee()
This is the constructor used for initializing the class settings and for bringing up the screen.

void actionPerformed(final ActionEvent e)
This function handles the GUI input. If submit is pressed the next screen is brought up based on the option.

int employeeCheck()
This function checks the employee field and grabs the employee id.

Sublclass: Employee

This class is responsible for either producing the Add Employee or Edit Employee window.

Employee(int employee_id)
The constructor used for building the Edit Employee window. The employee id must be supplied in order to grab the correct employee record from the database.

Employee()
This is the constructor used for building the Add Employee window.

void actionPerformed(final ActionEvent e)
The function which is used for handling the GUI input for both the Add Employee and Edit Employee windows. It handles the submit button and makes sure that all the fields contain correct values. Then the results are either added or committed to the database.

char userCheck()
This function is used to check if the username is blank, and that the user name is not being used. It is only called when adding an employee. It will report any errors which occur.

char passwordCheck()
This function is called to validate that the password field and confirm password field contain the same value. It will report any errors which occur.

char firstNameCheck()

This function is called to validate that the first name field is not blank. It will report any errors which occur.

char lastNameCheck()
This function is called to validate that the first name field is not blank. It will report any errors which occur.

char phone1Check()
This function is called to validate that the main phone number field is not blank. It will report any errors which occur.

Class: schedule_employeeJob

The schedule_employeeJob class contains the functions for managing the Employee Job GUI, Employee Job GUI input, and for calling the database functions to insert/update employee job records.
The following employee job record attributes can be manipulated from this interface: Active, Employee, and Job. If the record is being edited, then only the Active status can change.

schedule_EmployeeJob()
This serves as the constructor of the class. It first calls the findEmployeeJob class.

Subclass: findEmployeeJob

This class is responsible for building the initial screen for the Employee Job window. It contains a list of active employees, a list of active jobs, and the options list: Add Employee Job, Edit Employee Job.

findEmployeeJob()
This is the constructor used for creating the initial screen for the Employee Job window.

void actionPerformed(final ActionEvent e)
This is the function used for handling the GUI input for the screen.

char employeeCheck()
This function checks that the employee field is not blank. It also grabs the employee id from the selection.

char employeeJobCheck()
This function checks that the job field is not blank. It also grabs the job id from the selection.

Subclass: EmployeeJob

The employeeJob subclass is responsible for creating and handling the employeeJob add/edit screen.

EmployeeJob(int none)
This is the constructor used for creating the screen for Edit Employee Job window.

EmployeeJob()
This is the constructor used for creating the screen for Add Employee Job window.

void actionPerformed(final ActionEvent e)
This is the function used for handling the Employee Job GUI input. It also calls functions to check if the employee_id and job_id combination exists in the database.

char jobCheck()
This function checks to see if the job field is blank. If it is not blank then it check to see if the job exists in the system. If it exists, then it grabs the corresponding job_id in the database. It will report any errors which occur.

char employeeCheck()
This function checks to see if the employee field is blank. If it is not blank then it grabs the corresponding employe_id in the database. It will report any errors which occur.

Class: schedule_job

The schedule_job class contains the functions for managing the Job GUI, Job GUI input, and for calling the database functions to insert/update job records.
The following job record attributes can be manipulated from this interface: Active, and Job Name. If the record is being edited all fields can be modified. (Note: Job Id is hidden.)

schedule_Job()
The schedule_Job constructor is used to create and manage the schedule_Job GUI interface. It calls the findJob class to build the initial screen.

Subclass: findJob extends JPanel implements ActionListener

The findJob subclass is responsible for building and maintaining the Job GUI. It consists of a list of jobs and options to edit or add jobs.

findJob()
The function findJob is used for creating the initial Job screen GUI interface.

void actionPerformed(final ActionEvent e)
The function actionPerformed is used for handling the Job GUI input.

char jobCheck()
This function is used to check if a job exists in the system. If it does then the job_id is determined. It will report any errors which occur.

Subclass: Job extends JPanel implements ActionListener

This subclass is responsible for building and maintaing the Job GUI. It consists of the fields: Active and Job Name.

Job(int job_id)
This constructor is used to create the Edit a job screen.

Job()
This constructor is used to create the Add a job screen.

void actionPerformed(final ActionEvent e)
This function handles the GUI input for the job window.

char nameCheck()
This function checks if the job name field is blank. If it is not blank, it checks if the name exists in the system. The appropriate error messages are created.

Class: schedule_schedule

The schedule_schedule class contains the functions for creating the Schedule GUI, handling the Schedule GUI inputs, and calling the database functions to update and insert employee schedule records.
The following attributes can be modified:
Job, Start Time, End Time, Employee, and Active. The position and dateS can not be modified.

schedule_Schedule()
This is the constructor for the class which initially calls the findSchedule class to build the screen.

Subclass: findSchedule

This is the subclass responsible for building and maintaining the initial schedule screen.

findSchedule()
This is the constructor which builds the initial schedule screen which includes the date selection.

void actionPerformed(final ActionEvent e)
This function monitors the GUI input for the schedule window.

char dateCheck()
This function checks whether or not the inputted date is correct. It will report any errors which occur.

Subclass: Schedule

This function is responsible for adding/editing employee schedules. The fields include: Job, Start time, End time, Employee, and Active.

Schedule()
This is the constructor which builds and maintains the schedule window.

void actionPerformed(final ActionEvent e)
This is the function which monitors the GUI input for the schedule window.

Subclass class TableRender

This class is used for building and maintaining the schedule table.

TableRender()
This is the constructor used for building the schedule table.

void initColumnSizes(JTable table)
This function is used for initializing the column sizes for the schedule table.

void setUpEmployeeColumn(JTable table,TableColumn EmployeeColumn)
This function is used for building the employee column schedule table. It will build a list of active employees by accessing the database through the schedule_database class.

void setupJobColumn(JTable table, TableColumn JobColumn)
This function is used for building the Job column in the schedule table. It will build a list of active jobs by accessing the database through the schedule_database class.

void setupTimeColumn(JTable table, TableColumn C)
This function is used for building the start time and end time columns in the schedule table.

String checkIt(int row)
This function checks if there are any errors or inaccurate information in a table record. It will report any errors which occur.

checkScheduleRecord(int CMD)
This function checks if there are any errors or inaccuracies in the table. It will report any errors which occur.

int updateScheduleRecord()
This function updates or adds a schedule record.

Subclass MyTableModel

This subclass is used for building the table which can be organized in ascending order by column, and moved around.

int getColumnCount()
This function returns the number of columns.

String getColumnName(int col)
This function returns the column names.

Object getValueAt(int row, int col)
This function returns the value at a certain position in the table.

Class getColumnClass(int c)
This function returns the class of a column.

boolean isCellEditable(int row, int col)
This function returns if a column is editable (True/False).

void populateTable()
This function populates the table with a certain value. This is used at initialization.

setValueAt(Object value, final int row, int col, int C)
This function is used to set a particular position in the table with a particular value.

Class: schedule_vacation

The schedule_vacation class contains the functions for creating the Vacation GUI, handling the Vacation GUI inputs, and calling the database functions to update and insert employee vacation records.
The following attributes can be modified:
Employee, and Active. The position and dateS can not be modified.

schedule_Vacation()
This function is used for constructing and managing the vacation screens.

Subclass: findVacation

This function is used for creating the initial vacation screen.

findVacation()
This constructor is used for building and managing the initial vacation screen. The fields include date.

void actionPerformed(final ActionEvent e)
This function is used for handling the GUI input for the initial screen.

char dateCheck()
This function is used to check that the date is valid option. It will report any errors which occur.

Subclass: Vacation

This class is used for building and maintaining the vacation GUI.

Vacation()
This constructor is used for building the vacation GUI.

void actionPerformed(final ActionEvent e)
This function is used for handling the GUI input.

Subclass class TableRender

This class is used for building and maintaining the vacation table.

TableRender()
This is the constructor used for building the schedule table.

void initColumnSizes(JTable table)
This function is used for initializing the column sizes for the vacation table.

void setUpEmployeeColumn(JTable table,TableColumn EmployeeColumn)
This function is used for building the employee column vacation table. It accesses the database using the schedule_database class to build an active list of employees.

String checkIt(int row)
This function checks if there are any errors or inaccurate information in a table record. It will report any errors which occur.

checkVacationRecord(int CMD)
This function checks if there are any errors or inaccuracies in the table. It will report any errors which occur.

int updateVacationRecord()
This function updates or adds a vacation record.

Subclass MyTableModel

This subclass is used for building the table which can be organized in ascending order by column, and moved around.

int getColumnCount()
This function returns the number of columns.

String getColumnName(int col)
This function returns the column names.

Object getValueAt(int row, int col)
This function returns the value at a certain position in the table.

Class getColumnClass(int c)
This function returns the class of a column.

boolean isCellEditable(int row, int col)
This function returns if a column is editable (True/False).

void populateTable()
This function populates the table with a certain value. This is used at initialization.

setValueAt(Object value, final int row, int col, int C)
This function is used to set a particular position in the table with a particular value.

Class: schedule_printEmployee

The schedule_printEmployee class contains the functions for managing the Employee Report GUI, Employee Report GUI input, and for calling the database functions to view Employee data and Employee Schedule Data.
The interface allows a user to see all the work and vacation schedule records for an employee falling between the starting date and ending date. Also a user can print out the detailed employee information.

schedule_PrintEmployee()
This function is used to create and maintain the employee report screen.

Subclass: Schedule

This class is used for building and maintaining the initial screen for employee report.

schedule(int Admin)
This constructor is used for building the employee report screen.

void actionPerformed(final ActionEvent e)
This function is used for handling the employee report GUI input.

char employeeCheck()
This function is used to determine if an employee record exists in the system. If it does then it finds the employee_id. It will report any errors which occur.

char dateCheck()
This function is used to determine the date from the various date fields. It checks whether or not that the date is valid. It will report any errors which occur.

printEmployeeSchedule(int CMD)
This function is used to print the employee schedule/info. It utilizes the ShowText class.

Class: schedule_printStaff

The schedule_printStaff class contains the functions for managing the Staff Report GUI, Staff Report GUI input, and for calling the database functions to view Staff data.
This interface locates staff information and staff work schedules. The following options are allowed:

- Print Staff Schedule: All staff or only the staff capable of performing a particular job.
- Print Staff Available: All staff or only the staff capable of performing a particular job.
- Print Staff Info Active: All staff or only the staff capable of performing a particular job.
- Print Staff Info InActive: All staff or only the staff capable of performing a particular job.
- Print Staff Info: All staff or only the staff capable of performing a particular job.
- Print Staff Schedule Order by start time, job name, employee name: All staff or only the staff capable of performing a particular job.
- Print Staff Schedule Order by start time, employee name, job name: All staff or only the staff capable of performing a particular job.
- Print Staff Schedule Order by job name, start time, employee name: All staff or only the staff capable of performing a particular job.
- Print Staff Schedule Order by job name, employee name, start time: All staff or only the staff capable of performing a particular job.
- Print Staff Schedule Order by employee name , start time, job name: All staff or only the staff capable of performing a particular job.
- Print Staff Schedule Order by employee name , job name, start time: All staff or only the staff capable of performing a particular job.

schedule_PrintStaff(()
This constructor is used for building the staff report window.

Subclass: Schedule

This class is used for building and maintaining the Staff Report window.

schedule()
This function is used for building and maintaining the staff report GUI window.

void actionPerformed(final ActionEvent e)
This function is used for handling the GUI input for the Staff Report window.

char dateCheck()
This function checks the date for validity. It will report any errors which occur.

char employeeJobCheck()
This function checks the employee Job field. If it is not blank, then it determines the job_id.

printStaffSchedule(int CMD)
This function is used for printing the staff report to the screen. It calls the ShowText class to display the results.

Class: schedule_showTest

The schedule schedule_showTest class contains function which, check the database connection. It will return either a confirmation message or an error message.

Class: checkDate

The checkDate class contains functions used for compressing the date into an integer, decompressing the date into a String, constructing the date, and performing date checking.

int getYearInt(String date)
This function reads a date string and returns the year in integer format.

int getMonthInt(String date)
This function reads a date string and returns the month in integer format.

int getDayInt(String date)
This function reads a date string and returns the day in integer format.

String incrementDate(String date)
This function reads a date and increments it by one and returns the date.

int getDateInt(String date)
This function reads a date string and returns it in the integer format.

String getMonthString(String date)
This function reads a date string and returns the month name.

String makeDate(String monthString, String dayString, String yearString)
This function creates a date string given the month string, day string, and year string.

String translateMonth(String monthString)
This function translates a month name string into an integer string. (January = 01)

char check(String monthString, String dayString, String yearString)
This function checks if a month string, day string, and year string yields an acceptable date.

Class: checkFormat

The checkFormat class contains miscellaneous general purpose functions used throughout the program. For example there is function for extracting the employee_id from the string which contains the last name, first name, employee_id.

int getJobID(String name)
This function returns the job_id if given the name.

String getJobName(int jobID)
This function returns the job name if supplied the job_id

int getEmployeeID(String line)
This function returns the employee_id if given the employee last name, first name, employee_id.

String makeEmployeeLine(int employeeId)
This function returns the string for an employee_id: employee last name, first name, employee_id.

Class: checkTime

The checkTime class contains functions used for compressing the time into an integer, decompressing the time into a String, constructing the time, and performing time checking.

String StoragetoTimeString2(String time)
This function returns the time in the second system format.

String StoragetoTimeString(String time)
This function returns the time in the initial system format.

String TimeStringtoStorage(String time)
This function returns the time format in the system format to be stored in the database.

String toMilitaryTime(String hourString, String amPmString)
This function returns the hour and minute strings into military time format.

int getHourInt(String time)
This function returns the hour from a military time string.

int getMinInt(String time)
This function returns the minute from a military time string.

String getAmPmString(String time)
This function returns the am or pm from a military time string.

String makeTime(String hourString, String minuteString)
This function creates a time string out of the hour and minute string.

int makeTimeInt(String time)
This function creates an integer out the time string.

char check(String hourString, String minuteString)
This function checks if the time string is valid. It will report any errors which occur.

Class: cyclingSpinnerList

The cyclingSpinnerList class was obtained from the Java website. It handles a spinner list object.

CyclingSpinnerListModel()
This is the constructor for the class.

void setLinkedModel(SpinnerModel linkedModel)
This function returns the model.

Object getNextValue()
This function returns the next value.

Object getPreviousValue()
This function returns the previous model.

Class: help

The help class contains functions for creating the help screen.

help()
This is the help constructor.

void AddHelpLine(String Line)
This function is used to add a line to the help text window.

void actionPerformed(ActionEvent evt)
This function is used to handle actions in the help window.

void createAndShowHelp()
This function is used to show the help window.

Class: print2DPrinterJob

The print2DpritnerJob class contains functions for handling printer jobs.

Print2DPrinterJob()
This is the constructor for the print2DPritnerJob class.

void Print()
This function is used for creating the print job.

int print(Graphics g,PageFormat pf,int pageIndex)
This function is used for printing a particular page.

Class: record_employee

The record_employee class is an object used to transfer employee record data between different interfaces. It contains all the values used in the database record plus an error field.

Class: record_employeeJob

The record_employeeJob class is an object used to transfer employeeJob record data between different interfaces. It contains all the values used in the database record plus an error field.

Class: record_Job

The record_ob class is an object used to transfer Job record data between different interfaces. It contains all the values used in the database record plus an error field.

Class: record_Schedule

The record_schedule class is an object used to transfer employee schedule record data between different interfaces. It contains all the values used in the database record plus an error field.

Class: record_Vacation

The record_vacation class is an object used to transfer employee vacation record data between different interfaces. It contains all the values used in the database record plus an error field.

Class: SpringUtlilities

The SpringUtilities class was obtained from the Java website. It handles a spinner list object.

static void makeGrid(Container parent, int rows, int cols, int initialX, int initialY, int xPad, int yPad)
This function is used to setup the grid for the spring object.

void makeCompactGrid(Container parent, int rows, int col, int initialX, int initialY, int xPad, int yPad)
This function is used for finding the minimum size for the spring object grid.

Class: stringToInt

This stringToInt class contains miscellaneous functions for translating a string to an integer.

int power_any(int x, int n)
This function is used for finding X raised to the n power.

stringToInt(String string)
This function is used for translating a string to a hexadecimal integer value.

Class: tableSorter

The tableSorter class was obtained from the Java website and was slightly modified to only support sortting in assending order. Details can be found on the java.sun.com website.

Class: ShowText

The ShowText class contains the functions for handing the GUI interface employed in Staff Report and Employee Report after submit is pressed. It contains a Print button, Save button, Find All button, Find next button, a Search text field, and a text window.

Showtext(String filename1, int delete1)
This constructor is used for reading a file and displaying it to a screen. In addition there are buttons for sending a print job to a printer, performing a search, and saving to a file. The print2DPrinterJob class is used to handle print jobs.

void actionPerformed(ActionEvent e)
This function is used for handing the GUI input for the show text window. The find, find all, print, and save buttons are handled in this function.

void Search0()
This function is used to handle the Find button.

void Search1()
This function is used to handle the Find All button.

Search(int type)
This function is the main function for the various searches.