

coolgle  
1.2

Generated by Doxygen 1.8.5

Thu Dec 5 2013 06:18:53



# Contents

<b>1</b>	<b>Namespace Index</b>	<b>1</b>
1.1	Packages . . . . .	1
<b>2</b>	<b>Hierarchical Index</b>	<b>3</b>
2.1	Class Hierarchy . . . . .	3
<b>3</b>	<b>Class Index</b>	<b>5</b>
3.1	Class List . . . . .	5
<b>4</b>	<b>File Index</b>	<b>7</b>
4.1	File List . . . . .	7
<b>5</b>	<b>Namespace Documentation</b>	<b>9</b>
5.1	Package coolgle . . . . .	9
<b>6</b>	<b>Class Documentation</b>	<b>11</b>
6.1	coolgle.AddLocationGui Class Reference . . . . .	11
6.1.1	Detailed Description . . . . .	12
6.1.2	Constructor & Destructor Documentation . . . . .	12
6.1.2.1	AddLocationGui . . . . .	12
6.1.2.2	AddLocationGui . . . . .	12
6.1.2.3	AddLocationGui . . . . .	13
6.1.3	Member Function Documentation . . . . .	13
6.1.3.1	setUpGui . . . . .	13
6.2	coolgle.adminGui Class Reference . . . . .	14
6.2.1	Constructor & Destructor Documentation . . . . .	15
6.2.1.1	adminGui . . . . .	15
6.2.2	Member Function Documentation . . . . .	16
6.2.2.1	delUser . . . . .	16
6.2.2.2	editLocation . . . . .	17
6.2.2.3	logOut . . . . .	17
6.2.2.4	modUser . . . . .	18
6.2.2.5	populateLocationList . . . . .	18

6.2.2.6	populateUserList . . . . .	19
6.2.2.7	recreateDatabase . . . . .	19
6.3	coolgle.adminSwitchGui Class Reference . . . . .	20
6.3.1	Detailed Description . . . . .	21
6.3.2	Constructor & Destructor Documentation . . . . .	21
6.3.2.1	adminSwitchGui . . . . .	21
6.4	coolgle.Coord Class Reference . . . . .	21
6.4.1	Constructor & Destructor Documentation . . . . .	21
6.4.1.1	Coord . . . . .	21
6.4.1.2	Coord . . . . .	22
6.4.1.3	Coord . . . . .	22
6.4.2	Member Function Documentation . . . . .	22
6.4.2.1	getLatitude . . . . .	22
6.4.2.2	getLongitude . . . . .	22
6.4.2.3	setLatitude . . . . .	23
6.4.2.4	setLongitude . . . . .	23
6.4.2.5	toString . . . . .	23
6.5	coolgle.Driver Class Reference . . . . .	23
6.5.1	Member Function Documentation . . . . .	23
6.5.1.1	main . . . . .	23
6.6	coolgle.FileManager Class Reference . . . . .	23
6.6.1	Member Function Documentation . . . . .	24
6.6.1.1	addSearch . . . . .	24
6.6.1.2	addUserFile . . . . .	24
6.6.1.3	getPrev . . . . .	25
6.6.1.4	isFull . . . . .	26
6.6.1.5	removeUserFile . . . . .	27
6.6.1.6	userFileExists . . . . .	27
6.7	coolgle.Geocoder Class Reference . . . . .	28
6.7.1	Member Function Documentation . . . . .	28
6.7.1.1	createLocation . . . . .	28
6.8	coolgle.KmlCreator Class Reference . . . . .	30
6.8.1	Constructor & Destructor Documentation . . . . .	30
6.8.1.1	KmlCreator . . . . .	30
6.8.2	Member Function Documentation . . . . .	30
6.8.2.1	FillKml . . . . .	30
6.8.2.2	openKml . . . . .	31
6.8.2.3	printTime . . . . .	32
6.9	coolgle.Location Class Reference . . . . .	33
6.9.1	Constructor & Destructor Documentation . . . . .	33

6.9.1.1	Location	33
6.9.1.2	Location	34
6.9.1.3	Location	34
6.9.1.4	Location	34
6.9.1.5	Location	34
6.9.2	Member Function Documentation	35
6.9.2.1	distanceTo	35
6.9.2.2	fileToString	35
6.9.2.3	getAddress	36
6.9.2.4	getCity	37
6.9.2.5	getCoord	37
6.9.2.6	getLatitude	37
6.9.2.7	getLongitude	37
6.9.2.8	getName	38
6.9.2.9	getState	38
6.9.2.10	isSame	39
6.9.2.11	printToFile	39
6.9.2.12	printToFile	40
6.9.2.13	setAddress	41
6.9.2.14	setCity	41
6.9.2.15	setCoord	41
6.9.2.16	setLatitude	42
6.9.2.17	setLongitude	42
6.9.2.18	setName	42
6.9.2.19	setState	43
6.9.2.20	toString	43
6.10	coolgle.mainGui Class Reference	44
6.10.1	Constructor & Destructor Documentation	45
6.10.1.1	mainGui	45
6.10.2	Member Function Documentation	46
6.10.2.1	changeTrip	46
6.10.2.2	clearSearch	47
6.10.2.3	logOut	48
6.10.2.4	main	48
6.10.2.5	populateLocationList	49
6.10.2.6	prepareTrip	50
6.10.2.7	removeSelectedMidTrip	51
6.10.2.8	updateEnd	52
6.10.2.9	updateMid	52
6.10.2.10	updateStart	53

6.10.2.11	updateStartingTime	53
6.10.2.12	updateVisualDisplay	54
6.10.2.13	validateTrip	55
6.10.2.14	viewHelp	56
6.10.2.15	viewInformation	56
6.11	coolgle.modifyUsersGui Class Reference	57
6.11.1	Constructor & Destructor Documentation	57
6.11.1.1	modifyUsersGui	57
6.11.2	Member Function Documentation	58
6.11.2.1	main	58
6.12	coolgle.Search Class Reference	59
6.12.1	Constructor & Destructor Documentation	60
6.12.1.1	Search	60
6.12.1.2	Search	60
6.12.1.3	Search	60
6.12.1.4	Search	61
6.12.2	Member Function Documentation	61
6.12.2.1	addLocation	61
6.12.2.2	clearData	61
6.12.2.3	fileToString	61
6.12.2.4	getEnd	62
6.12.2.5	getMidLocations	62
6.12.2.6	getNumMidLocations	63
6.12.2.7	getStart	63
6.12.2.8	getStartTime	63
6.12.2.9	getUserName	64
6.12.2.10	hasShortest	64
6.12.2.11	isInSearch	64
6.12.2.12	removeAtPosition	65
6.12.2.13	removeLocation	65
6.12.2.14	setDistance	66
6.12.2.15	setEnd	66
6.12.2.16	setLocations	66
6.12.2.17	setOptimized	66
6.12.2.18	setShortestPath	66
6.12.2.19	setStart	66
6.12.2.20	setStartTime	67
6.12.2.21	setUserName	67
6.12.2.22	toString	67
6.13	coolgle.SearchManager Class Reference	67

6.13.1	Member Function Documentation	68
6.13.1.1	sortShortestDistance	68
6.14	coolgle.SignInGui Class Reference	69
6.14.1	Constructor & Destructor Documentation	69
6.14.1.1	SignInGui	69
6.15	coolgle.SignUpGui Class Reference	70
6.15.1	Constructor & Destructor Documentation	70
6.15.1.1	SignUpGui	70
6.16	coolgle.UIManager Class Reference	71
6.16.1	Constructor & Destructor Documentation	71
6.16.1.1	UIManager	71
6.17	coolgle.UserAuthentication Class Reference	71
6.17.1	Member Function Documentation	71
6.17.1.1	addUser	71
6.17.1.2	addUserWithType	72
6.17.1.3	getEmail	73
6.17.1.4	getNames	73
6.17.1.5	getPassword	74
6.17.1.6	getType	75
6.17.1.7	logUser	75
6.17.1.8	removeUser	76
6.17.1.9	updateUser	77
6.17.1.10	userExists	78
6.18	coolgle.userSwitchGui Class Reference	79
6.18.1	Constructor & Destructor Documentation	79
6.18.1.1	userSwitchGui	79
<b>7</b>	<b>File Documentation</b>	<b>81</b>
7.1	src/coolgle/AddLocationGui.java File Reference	81
7.2	src/coolgle/adminGui.java File Reference	81
7.3	src/coolgle/adminSwitchGui.java File Reference	81
7.4	src/coolgle/Coord.java File Reference	81
7.5	src/coolgle/Driver.java File Reference	82
7.6	src/coolgle/FileManager.java File Reference	82
7.7	src/coolgle/Geocoder.java File Reference	82
7.8	src/coolgle/KmlCreator.java File Reference	82
7.9	src/coolgle/Location.java File Reference	83
7.10	src/coolgle/mainGui.java File Reference	83
7.11	src/coolgle/modifyUsersGui.java File Reference	83
7.12	src/coolgle/Search.java File Reference	83

7.13 src/coolgle/SearchManager.java File Reference . . . . .	83
7.14 src/coolgle/SignInGui.java File Reference . . . . .	84
7.15 src/coolgle/SignUpGui.java File Reference . . . . .	84
7.16 src/coolgle/UiManager.java File Reference . . . . .	84
7.17 src/coolgle/UserAuthentication.java File Reference . . . . .	84
7.18 src/coolgle/userSwitchGui.java File Reference . . . . .	85

<b>Index</b>	<b>86</b>
--------------	-----------



# Chapter 1

## Namespace Index

### 1.1 Packages

Here are the packages with brief descriptions (if available):

<a href="#">coolgle</a> . . . . .	9
-----------------------------------	---



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

coolgle.Coord . . . . .	21
coolgle.Driver . . . . .	23
coolgle.FileManager . . . . .	23
coolgle.Geocoder . . . . .	28
JFrame	
coolgle.AddLocationGui . . . . .	11
coolgle.adminGui . . . . .	14
coolgle.adminSwitchGui . . . . .	20
coolgle.mainGui . . . . .	44
coolgle.modifyUsersGui . . . . .	57
coolgle.SignInGui . . . . .	69
coolgle.SignUpGui . . . . .	70
coolgle.userSwitchGui . . . . .	79
coolgle.KmlCreator . . . . .	30
coolgle.Location . . . . .	33
coolgle.Search . . . . .	59
coolgle.SearchManager . . . . .	67
coolgle.UIManager . . . . .	71
coolgle.UserAuthentication . . . . .	71



## Chapter 3

# Class Index

### 3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">coolgle.AddLocationGui</a>	11
<a href="#">coolgle.adminGui</a>	14
<a href="#">coolgle.adminSwitchGui</a>	20
<a href="#">coolgle.Coord</a>	21
<a href="#">coolgle.Driver</a>	23
<a href="#">coolgle.FileManager</a>	23
<a href="#">coolgle.Geocoder</a>	28
<a href="#">coolgle.KmlCreator</a>	30
<a href="#">coolgle.Location</a>	33
<a href="#">coolgle.mainGui</a>	44
<a href="#">coolgle.modifyUsersGui</a>	57
<a href="#">coolgle.Search</a>	59
<a href="#">coolgle.SearchManager</a>	67
<a href="#">coolgle.SignInGui</a>	69
<a href="#">coolgle.SignUpGui</a>	70
<a href="#">coolgle.UIManager</a>	71
<a href="#">coolgle.UserAuthentication</a>	71
<a href="#">coolgle.userSwitchGui</a>	79



## Chapter 4

# File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

src/coolgle/AddLocationGui.java	81
src/coolgle/adminGui.java	81
src/coolgle/adminSwitchGui.java	81
src/coolgle/Coord.java	81
src/coolgle/Driver.java	82
src/coolgle/FileManager.java	82
src/coolgle/Geocoder.java	82
src/coolgle/KmlCreator.java	82
src/coolgle/Location.java	83
src/coolgle/mainGui.java	83
src/coolgle/modifyUsersGui.java	83
src/coolgle/Search.java	83
src/coolgle/SearchManager.java	83
src/coolgle/SignInGui.java	84
src/coolgle/SignUpGui.java	84
src/coolgle/UIManager.java	84
src/coolgle/UserAuthentication.java	84
src/coolgle/userSwitchGui.java	85





## Chapter 5

# Namespace Documentation

### 5.1 Package coolgle

#### Classes

- class [AddLocationGui](#)
- class [adminGui](#)
- class [adminSwitchGui](#)
- class [Coord](#)
- class [Driver](#)
- class [FileManager](#)
- class [Geocoder](#)
- class [KmlCreator](#)
- class [Location](#)
- class [mainGui](#)
- class [modifyUsersGui](#)
- class [Search](#)
- class [SearchManager](#)
- class [SignInGui](#)
- class [SignUpGui](#)
- class [UIManager](#)
- class [UserAuthentication](#)
- class [userSwitchGui](#)

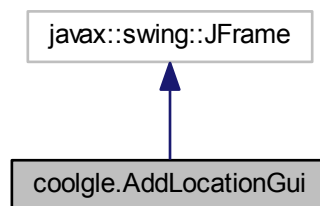


## Chapter 6

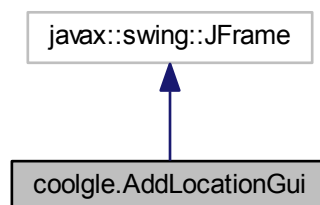
# Class Documentation

### 6.1 coolgle.AddLocationGui Class Reference

Inheritance diagram for coolgle.AddLocationGui:



Collaboration diagram for coolgle.AddLocationGui:



#### Public Member Functions

- [AddLocationGui](#) ([adminGui](#) mainFrame)
- [AddLocationGui](#) ([Location](#) newLocation, [adminGui](#) mainFrame)

- [AddLocationGui](#) ([Location](#) newLocation)
- void [setUpGui](#) ()

### 6.1.1 Detailed Description

constructor

### 6.1.2 Constructor & Destructor Documentation

#### 6.1.2.1 coolgle.AddLocationGui.AddLocationGui ( adminGui *mainFrame* )

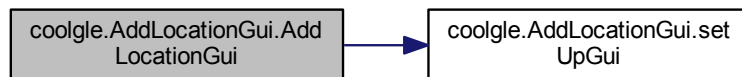
Creates new form AddLocation pre post adds location to filesystem and gui

```

37     {
38         setUpGui ();
39         status = 0;
40         m_main = mainFrame;
41     }

```

Here is the call graph for this function:



#### 6.1.2.2 coolgle.AddLocationGui.AddLocationGui ( *Location* newLocation, adminGui *mainFrame* )

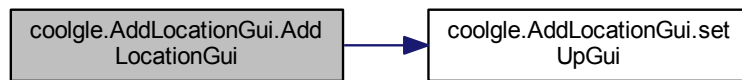
This will be used with the modify location screen, Creates new form AddLocation pre admin gui is set up post load the add location screen with the data already pre-filled in.

```

50     {
51         setUpGui ();
52
53         this.loNameText.setText(newLocation.getName());
54         this.loLatText.setText(String.valueOf(newLocation.getLatitude()));
55         this.loLongText.setText(String.valueOf(newLocation.getLongitude()));
56         this.loCityText.setText(newLocation.getCity());
57         this.loStateText.setText(newLocation.getState());
58         this.loAddressText.setText(newLocation.getAddress());
59
60         // No changing the latitude and longitude
61         this.loLatText.setEditable(false);
62         this.loLongText.setEditable(false);
63
64         status = 1;
65         m_main = mainFrame;
66     }

```

Here is the call graph for this function:



### 6.1.2.3 coolgle.AddLocationGui.AddLocationGui ( Location *newLocation* )

Called from [mainGui](#) page, only displays information, no editing. pre newlocation valid post adds location to filesystem

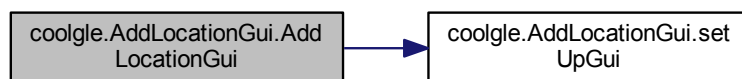
#### Parameters

<i>newLocation</i>	
--------------------	--

```

76     {
77         setUpGui ();
78
79         this.loNameText.setText (newLocation.getName ());
80         this.loLatText.setText (String.valueOf (newLocation.getLatitude ());
81         this.loLongText.setText (String.valueOf (newLocation.getLongitude ());
82         this.loCityText.setText (newLocation.getCity ());
83         this.loStateText.setText (newLocation.getState ());
84         this.loAddressText.setText (newLocation.getAddress ());
85
86         this.loNameText.setEditable (false);
87         this.loLatText.setEditable (false);
88         this.loLongText.setEditable (false);
89         this.loCityText.setEditable (false);
90         this.loStateText.setEditable (false);
91         this.loAddressText.setEditable (false);
92
93         saveLocationBtn.setText ("OK");
94
95         status = 2;
96     }
  
```

Here is the call graph for this function:



## 6.1.3 Member Function Documentation

### 6.1.3.1 void coolgle.AddLocationGui.setUpGui ( )

pre post sets up gui screen

```

104     {
105         Dimension screen = Toolkit.getDefaultToolkit ().getScreenSize ();
  
```

```
106         Dimension frm = super.getSize();
107         int xpos = (int) (screen.getWidth() / 8 - frm.getWidth() / 2);
108         int ypos = (int) (screen.getHeight() / 8 - frm.getHeight() / 2);
109         super.setLocation(xpos, ypos);
110         initComponents();
111
112         // Makes it so that you can exit from pop up messages and this window without closing the entire
113         app this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
114     }
```

Here is the caller graph for this function:

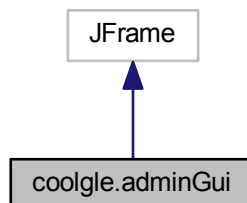


The documentation for this class was generated from the following file:

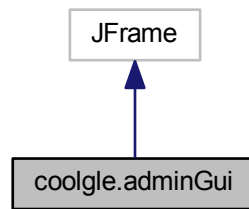
- `src/coolgle/AddLocationGui.java`

## 6.2 coolgle.adminGui Class Reference

Inheritance diagram for `coolgle.adminGui`:



Collaboration diagram for coolgle.adminGui:



## Public Member Functions

- `adminGui` (String userName)
- void `logOut` ()
- void `populateUserList` ()
- void `populateLocationList` ()
- void `editLocation` (Location newLoc)
- void `recreateDatabase` ()
- void `modUser` ()
- void `delUser` ()

## 6.2.1 Constructor & Destructor Documentation

### 6.2.1.1 coolgle.adminGui.adminGui ( String userName )

constructor Creates new form GuiUI

```

43     {
44         Dimension screen = Toolkit.getDefaultToolkit().getScreenSize();
45         Dimension frm = super.getSize();
46         int xpos = (int) (screen.getWidth() / 8 - frm.getWidth() / 2);
47         int ypos = (int) (screen.getHeight() / 30);
48         super.setLocation(xpos, ypos);
49         initComponents();
50
51         user = userName;
52
53         // Populate the location list
54         populateLocationList();
55         populateUserList();
56
57         //Allows the logout or close popup to appear when closing window
58         this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
59         this.addWindowListener(new java.awt.event.WindowAdapter()
60         {
61             @Override
62             public void windowClosing(java.awt.event.WindowEvent windowEvent)
63             {
64                 Object[] options = {"Log out","Exit"};
65                 int n = JOptionPane.showOptionDialog(null,
66                     "Exit?", "Exit",JOptionPane.YES_NO_CANCEL_OPTION,
67                     JOptionPane.QUESTION_MESSAGE,
68                     null,
69                     options,
70                     options[1]);
71
72                 if (n == 1)
73                 {
74                     System.exit(0);
75                 }
76             }
77         });
78     }
79 }

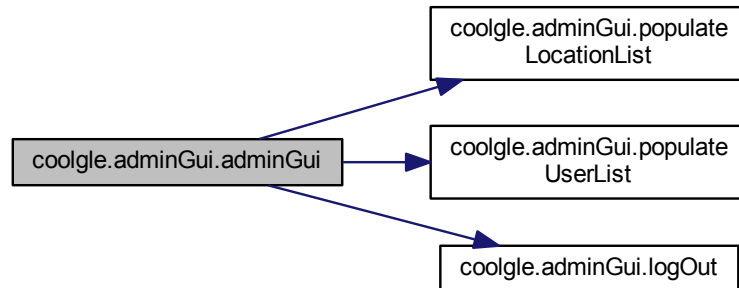
```

```

76         else // Log out
77         {
78             logout ();
79         }
80     }));
81 }

```

Here is the call graph for this function:



## 6.2.2 Member Function Documentation

### 6.2.2.1 void coolgle.adminGui.delUser ( )

`delUser` Description - Deletes the selected user. Shows warning in the following cases :

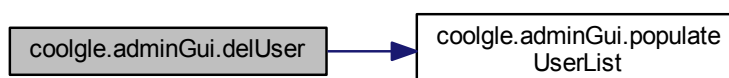
- Always (Asks if admin is sure they want to delete)
- Nothing is selected

```

552     {
553         if (userDisplay.getSelectedIndex() != -1)
554         {
555             int confirm = JOptionPane.showConfirmDialog(null, "Are you sure you want to delete \n \" "
556                                                         + userDisplay.getSelectedValue().toString() + " \"
557         " );
558             if (confirm == 0)
559             {
560                 UserAuthentication delUser = new UserAuthentication();
561                 delUser.removeUser(userDisplay.getSelectedValue().toString(), true); //Remove file as well
562                 populateUserList();
563             }
564         else // No user selected
565         {
566             JOptionPane.showMessageDialog(null, "No user selected", "Error", JOptionPane.ERROR_MESSAGE);
567         }
568     }

```

Here is the call graph for this function:





## 6.2.2.2 void coolgle.adminGui.editLocation ( Location newLoc )

Description- Replaces the currently selected location with a new one Called from modify button -> save

Parameters

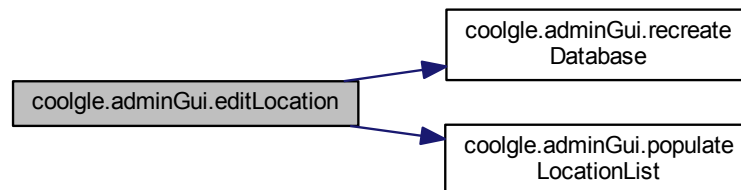
<i>newLoc</i>	
---------------	--

```

484  {
485      // Replace currently selected object in list with new location
486      //(TODO : maybe put this in a copy function in location class)
487      locationlist.get( locationDisplay.getSelectedIndex() ).setAddress(newLoc.getAddress());
488      locationlist.get( locationDisplay.getSelectedIndex() ).setLatitude(newLoc.getLatitude());
489      locationlist.get( locationDisplay.getSelectedIndex() ).setLongitude(newLoc.getLongitude());
490      locationlist.get( locationDisplay.getSelectedIndex() ).setCity(newLoc.getCity());
491      locationlist.get( locationDisplay.getSelectedIndex() ).setState(newLoc.getState());
492      locationlist.get( locationDisplay.getSelectedIndex() ).setName(newLoc.getName());
493
494      // Recreate database baesd on new arraylist
495      recreateDatabase();
496
497      // Repopulate list
498      populateLocationList();
499  }

```

Here is the call graph for this function:



## 6.2.2.3 void coolgle.adminGui.logOut ( )

logOut Description - To be called when user exits screen, asks if user wants to log out or completely close

```

100  {
101      SignInGui logOut = new SignInGui();
102      logOut.setVisible(true);
103      this.setVisible(false);
104  }

```

Here is the caller graph for this function:



#### 6.2.2.4 void coolgle.adminGui.modUser ( )

modUser Description - Launches the addLocation dialog with the currently selected item loaded in

```

532     {
533         if (userDisplay.getSelectedIndex() != -1)
534         {
535             modifyUsersGui modUser = new modifyUsersGui( this, userDisplay.getSelectedValue()
536             .toString() );
537             modUser.setVisible(true);
538         }
539         else // No user selected
540         {
541             JOptionPane.showMessageDialog(null, "No user selected", "Error", JOptionPane.ERROR_MESSAGE);
542         }
543     }

```

#### 6.2.2.5 void coolgle.adminGui.populateLocationList ( )

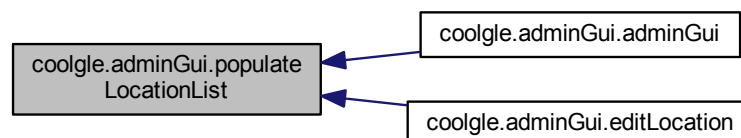
populateLocationList Description - Called whenever there is a change to the locationDatabase file. Refreshes the location list to match the newest changes

```

444     {
445         try
446         {
447             // Clear out current list
448             locationlist.clear();
449
450             // Fill in Location List
451             DefaultListModel listModel;
452             listModel = new DefaultListModel();
453             locationDisplay.setModel(listModel);
454             BufferedReader br;
455             try
456             {
457                 br = new BufferedReader(new FileReader("LocationDatabase.txt"));
458                 String locationString;
459                 //ArrayList<Location> DatabaseLocations = new ArrayList<Location>();
460                 // Read file Location by location
461                 while ((locationString = br.readLine()) != null)
462                 {
463                     Location newLocation = new Location(locationString);
464                     locationlist.add(newLocation);
465                     listModel.addElement(newLocation.getName());
466                 }
467             }
468             catch (FileNotFoundException ex)
469             {
470                 JOptionPane.showMessageDialog(null, "Admin - Could not find database file", "Failure",
471                 JOptionPane.ERROR_MESSAGE);
472             }
473             catch (IOException ex)
474             {
475                 JOptionPane.showMessageDialog(null, "Admin - Error opening database file", "Failure",
476                 JOptionPane.ERROR_MESSAGE);
477             }
478         }
479     }

```

Here is the caller graph for this function:



## 6.2.2.6 void coolgle.adminGui.populateUserList ( )

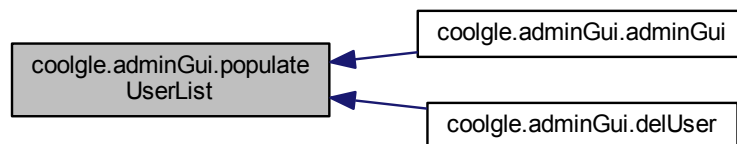
populateUserList Description - Called whenever there is a change to the UID-pass file. Refreshes the user list to match the newest changes

```

423     {
424         // Fill in User List
425         DefaultListModel listModel = new DefaultListModel();
426         userDisplay.setModel(new DefaultListModel());
427
428         UserAuthentication userNames = new UserAuthentication();
429         ArrayList<String> allNames = userNames.getNames();
430
431         for ( int i = 0; i < allNames.size(); i++ )
432         {
433             listModel.addElement( allNames.get(i) );
434         }
435     }

```

Here is the caller graph for this function:



## 6.2.2.7 void coolgle.adminGui.recreateDatabase ( )

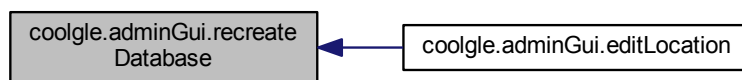
recreateDatabase Description - Syncs the locationDatabase.txt file with the data in the display. Called whenever there is a change in the display.

```

507     {
508         // Clear old database file
509         PrintWriter writer;
510         try
511         {
512             writer = new PrintWriter("LocationDatabase.txt");
513             writer.print("");
514             writer.close();
515
516             // Repopulate database file from arraylist
517             for (int i = 0; i < locationlist.size(); i++)
518                 locationlist.get(i).printToFile("LocationDatabase.txt");
519
520         }
521         catch (FileNotFoundException ex)
522         {
523             JOptionPane.showMessageDialog(null, "Admin - Database not found", "Failure",
524                                         JOptionPane.ERROR_MESSAGE);
525         }
526     }

```

Here is the caller graph for this function:

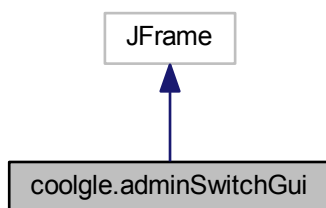


The documentation for this class was generated from the following file:

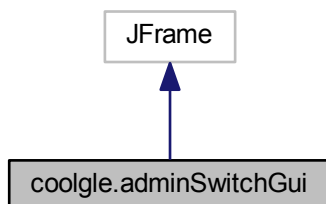
- `src/coolgle/adminGui.java`

### 6.3 coolgle.adminSwitchGui Class Reference

Inheritance diagram for `coolgle.adminSwitchGui`:



Collaboration diagram for `coolgle.adminSwitchGui`:



#### Public Member Functions

- `adminSwitchGui` (`mainGui` `m_main`, `String` `userName`)

### 6.3.1 Detailed Description

Author

rdg77\_000

### 6.3.2 Constructor & Destructor Documentation

#### 6.3.2.1 coolgle.adminSwitchGui.adminSwitchGui ( mainGui *m\_main*, String *userName* )

Constructor Creates new form adminSwitch

```

33     {
34         Dimension screen = Toolkit.getDefaultToolkit().getScreenSize();
35         Dimension frm = super.getSize();
36         int xpos = (int) (screen.getWidth() / 8 - frm.getWidth() / 2);
37         int ypos = (int) (screen.getHeight() / 8 - frm.getHeight() / 2);
38         super.setLocation(xpos, ypos);
39
40         // Makes it so that you can exit from pop up messages and this window without closing the entire
41         app this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
42
43         user = userName;
44         mainFrame = m_main;
45
46         initComponents();
47     }

```

The documentation for this class was generated from the following file:

- [src/coolgle/adminSwitchGui.java](#)

## 6.4 coolgle.Coord Class Reference

### Public Member Functions

- [Coord](#) ()
- [Coord](#) (double latitude, double longitude)
- [Coord](#) (String latitude, String longitude)
- double [getLatitude](#) ()
- double [getLongitude](#) ()
- void [setLatitude](#) (double newLat)
- void [setLongitude](#) (double newLong)
- String [toString](#) ()

### 6.4.1 Constructor & Destructor Documentation

#### 6.4.1.1 coolgle.Coord.Coord ( )

Default Constructor for coord

```

24     {
25         latitude = 0;
26         longitude = 0;
27     }

```

#### 6.4.1.2 coolgle.Coord.Coord ( double *latitude*, double *longitude* )

Initialized Constructor for coord

```

35    {
36        this.latitude = latitude;
37        this.longitude = longitude;
38    }

```

#### 6.4.1.3 coolgle.Coord.Coord ( String *latitude*, String *longitude* )

Parsed Constructor for coord

```

43    {
44        this.latitude = Double.parseDouble(latitude);
45        this.longitude = Double.parseDouble(longitude);
46    }

```

### 6.4.2 Member Function Documentation

#### 6.4.2.1 double coolgle.Coord.getLatitude ( )

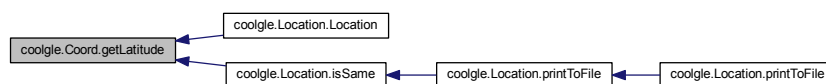
Description- getter for latitude

```

51    {
52        return latitude;
53    }

```

Here is the caller graph for this function:



#### 6.4.2.2 double coolgle.Coord.getLongitude ( )

Description- getter for longitude

```

58    {
59        return longitude;
60    }

```

Here is the caller graph for this function:



#### 6.4.2.3 void coolgle.Coord.setLatitude ( double *newLat* )

Description- setter for latitude

```
65      {  
66          latitude = newLat;  
67      }
```

#### 6.4.2.4 void coolgle.Coord.setLongitude ( double *newLong* )

Description- setter for longitude

```
72      {  
73          longitude = newLong;  
74      }
```

#### 6.4.2.5 String coolgle.Coord.toString ( )

Description- overridden text file for toString method

```
80      {  
81          String msg = String.format("%.05f", latitude) + " " + String.format("%.05f", longitude);  
82          return msg;  
83      }
```

The documentation for this class was generated from the following file:

- src/coolgle/[Coord.java](#)

## 6.5 coolgle.Driver Class Reference

### Static Public Member Functions

- static void [main](#) (String args[])

#### 6.5.1 Member Function Documentation

##### 6.5.1.1 static void coolgle.Driver.main ( String *args*[] ) [static]

```
20      {  
21          UIManager UI = new UIManager();  
22      }
```

The documentation for this class was generated from the following file:

- src/coolgle/[Driver.java](#)

## 6.6 coolgle.FileManager Class Reference

### Public Member Functions

- [Search getPrev](#) (String userName, int searchIndex)
- boolean [isFull](#) (File file)
- void [addSearch](#) ([Search](#) search)
- void [removeUserFile](#) (String userName)
- boolean [userFileExists](#) (String userName)
- void [addUserFile](#) (String userName)

## 6.6.1 Member Function Documentation

### 6.6.1.1 void coolgle.FileManager.addSearch ( Search search )

addSearch(SEARCH search) Description- SEARCH to search.txt (add database entry) Pre- user must exist Post- adds a valid completed search to users history

```

165     {
166         try
167         {
168             String temp = PATH + search.getUserName() + SUFFIX;
169             File file = new File(temp);
170
171             if( file.exists() )
172             {
173                 if( isFull(file) )
174                 {
175                     JOptionPane.showMessageDialog(null, "File is full, cannot save search"
176                     , "Failure", JOptionPane.ERROR_MESSAGE);
177                 }
178                 else
179                 {
180                     //append data to end of file
181                     PrintWriter out = new PrintWriter(new BufferedWriter(new FileWriter(file, true)));
182
183                     out.write(search.fileToString().replaceAll("\n", System.lineSeparator()));
184
185                     out.close();
186                 }
187             }
188             else
189             {
190                 JOptionPane.showMessageDialog(null, "User (" + search.getUserName() + ") file not found,
191                 Cannot save search"
192                 , "Failure", JOptionPane.ERROR_MESSAGE);
193             }
194         }
195         catch (IOException e)
196         {
197             JOptionPane.showMessageDialog(null, "IOException when opening file, search will not be saved"
198             , "Failure", JOptionPane.ERROR_MESSAGE);
199         }
200     }

```

Here is the call graph for this function:



### 6.6.1.2 void coolgle.FileManager.addUserFile ( String userName )

addUser() Description- adds file for users search Pre- User does not exist Post-adds text file for user

```

243     {
244         //adds new user SEARCH file
245         String temp = "";
246         temp += PATH + userName + SUFFIX;
247         File file = new File(temp);
248         try
249         {
250             if(file.exists())
251             {
252                 JOptionPane.showMessageDialog(null, "File Already Exists", "Failure",

```

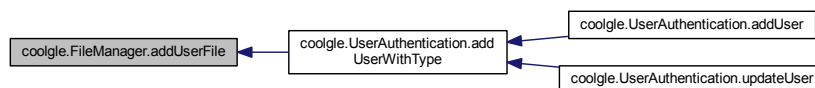


```

        JOptionPane.ERROR_MESSAGE);
253     }
254     else
255     {
256         file.createNewFile();
257     }
258 }
259 catch (IOException e)
260 {
261     JOptionPane.showMessageDialog(null, "IOException Error", "Failure", JOptionPane.ERROR_MESSAGE);
262 }
263 }

```

Here is the caller graph for this function:



### 6.6.1.3 Search coolgle.FileManager.getPrev ( String userName, int searchIndex )

constructor for FileManager object

```

public FileManager()
{

```

Nothing } pre must have valid text file post prevSearch is returned from user textFile

#### Parameters

<i>userName</i>	
<i>searchIndex</i>	

#### Returns

```

41 {
42     Search prevSearch = new Search();
43     try
44     {
45         String userPath = PATH + userName + SUFFIX;
46         String newLine = "StartTime "; //This is the indicator that we've reached a new
search
47         String endMidLocs = "***endSearch"; //This is the indicator that we've reached the end of a
search
48         File file = new File(userPath);
49
50         ArrayList<Location> locations = new ArrayList<Location>();
51
52         if( file.exists() )
53         {
54             BufferedReader br = new BufferedReader(new FileReader(userPath));
55             //get all lines from file
56             int index = 0;
57             String currLine;
58             while ((currLine = br.readLine()) != null)
59             {
60                 //align scanner to searchIndex
61                 if( currLine.startsWith(newLine) ) // If start of a search
62                 {
63                     index++;
64                     if(index == searchIndex) //If this is the right search
65                     {
66                         String startingTime = currLine.substring(newLine.length());
67                         prevSearch.setStartTime(Double.valueOf(startingTime));
68

```

```

69         //Then as long as the next line isn't the end of search line, make locations
70         boolean hasMoreLocation = true;
71         while (hasMoreLocation)
72         {
73             currLine = br.readLine();
74             if (currLine.compareTo(endMidLocs) == 0)
75                 hasMoreLocation = false;
76             else //Its a location
77             {
78                 Location newLocation = new Location(currLine);
79                 locations.add(newLocation);
80             }
81         }
82         prevSearch.setStart(locations.get(0));
83         locations.remove(0);
84         prevSearch.setEnd(locations.get(locations.size() - 1));
85         locations.remove(locations.size() - 1);
86         for (int i = 0; i < locations.size(); i++)
87         {
88             prevSearch.addLocation(locations.get(i));
89         }
90     }
91 }
92 }
93 br.close();
94 }
95 else
96 {
97     JOptionPane.showMessageDialog(null, "Get Previous - File Not Found \n Could not access "
98         + userName + " Search " + searchIndex
99         , "Failure", JOptionPane.ERROR_MESSAGE);
100 }
101 }
102 }
103 catch (IOException e)
104 {
105     JOptionPane.showMessageDialog(null, "File IOException Error"
106         , "Failure", JOptionPane.ERROR_MESSAGE);
107 }
108
109     prevSearch.setUserName(userName);
110     prevSearch.setOptimized(false);
111
112     return prevSearch;
113 }

```

#### 6.6.1.4 boolean coolgle.FileManager.isFull ( File file )

Description- need non empty file, shows if file has MAX\_PREVIOUS\_TRIPS searches (full) or less return true if file is full

```

121 {
122     int searchCount = 0;
123
124     String newLine = "StartTime";
125     try {
126         Scanner line = new Scanner(file);
127         Scanner inLine = null ;
128         //get all lines from file
129         while(line.hasNext())
130         {
131             String currLine = line.nextLine();
132             inLine = new Scanner(currLine);
133             //find new search token
134             if(inLine.next().equals(newLine))
135             {
136                 searchCount +=1;
137                 if(searchCount >= MAX_PREVIOUS_TRIPS)
138                 {
139                     line.close();
140                     inLine.close();
141                     return true;
142                 }
143             }
144         }
145         line.close();
146         inLine.close();
147     }
148     catch (FileNotFoundException e)
149     {
150         JOptionPane.showMessageDialog(null, "IsFull - File Not Found Exception"
151             , "Failure", JOptionPane.ERROR_MESSAGE);

```

```

152     }
153
154     return false;
155 }

```

Here is the caller graph for this function:



#### 6.6.1.5 void coolgle.FileManager.removeUserFile ( String userName )

`removeUser()` Description- removes users searches Pre- user exists Post- removes text file for user

```

208 {
209     //REMOVES new user SEARCH file
210     String temp = "";
211     temp += PATH + userName + SUFFIX;
212     File file = new File(temp);
213     if( !file.exists() )
214     {
215         JOptionPane.showMessageDialog(null, "User File Doesn't Exist", "Failure",
JOptionPane.ERROR_MESSAGE);
216     }
217     else
218     {
219         file.deleteOnExit();
220     }
221 }

```

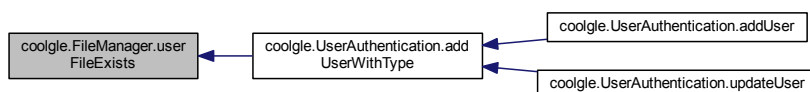
#### 6.6.1.6 boolean coolgle.FileManager.userFileExists ( String userName )

```

225 {
226     boolean findFound = false;
227     String temp = "";
228     temp += PATH + userName + SUFFIX;
229     File file = new File(temp);
230     if( file.exists() )
231     {
232         findFound = true;
233     }
234     return findFound;
235 }

```

Here is the caller graph for this function:



The documentation for this class was generated from the following file:

- `src/coolgle/FileManager.java`

## 6.7 coolgle.Geocoder Class Reference

### Public Member Functions

- [Location](#) [createLocation](#) (String name)

#### 6.7.1 Member Function Documentation

##### 6.7.1.1 Location coolgle.Geocoder.createLocation ( String name )

[createLocation](#) Description - Creates and returns a location object based on a input string Parameters - String, what to search in googles geocoder; Ex : "Golden Gate Bridge" Return - [Location](#) object, all fields filled in.

```

53     {
54         Location location = new Location();
55         try
56         {
57             // URL prefix to the geocoder
58             final String GEOCODER_REQUEST_PREFIX_FOR_XML= "http://maps.google.com/maps/api/geocode/xml";
59
60             // prepare a URL to the geocoder
61             URL url = new URL(GEOCODER_REQUEST_PREFIX_FOR_XML + "?address=" + URLEncoder.encode(name, "
UTF-8") + "&sensor=false");
62
63             // prepare an HTTP connection to the geocoder
64             HttpURLConnection conn = (HttpURLConnection) url.openConnection();
65
66             Document geocoderResultDocument = null;
67
68             //try
69             // open the connection and get results as InputSource.
70             conn.connect();
71             InputSource geocoderResultInputSource = new InputSource(conn.getInputStream());
72
73             // read result and parse into XML Document
74             geocoderResultDocument = DocumentBuilderFactory.newInstance().newDocumentBuilder().parse(
geocoderResultInputSource);
75
76             //finally
77             conn.disconnect();
78
79             // prepare XPath
80             XPath xpath = XPathFactory.newInstance().newXPath();
81
82             // extract the result
83             NodeList resultNodeList = null;
84             resultNodeList = (NodeList) xpath.evaluate("/GeocodeResponse/result/formatted_address",
geocoderResultDocument, XPathConstants.NODESET);
85
86             if (resultNodeList.getLength() == 0)
87             {
88                 // Err : No location Found from search term
89                 // Just return a empty location, which can be handled later
90             }
91             else
92             {
93                 // Get Name Of Location
94                 resultNodeList = (NodeList) xpath.evaluate("
/GeocodeResponse/result[1]/address_component/short_name", geocoderResultDocument, XPathConstants.NODESET);
95                 if (resultNodeList.item(0) != null)
96                     location.setName(resultNodeList.item(0).getTextContent());
97
98                 String address = "";
99                 // Get Street Number Of Location
100                 resultNodeList = (NodeList) xpath.evaluate("
/GeocodeResponse/result[1]/address_component[type/text()='street_number']/long_name", geocoderResultDocument, XPathConst
101                 if (resultNodeList.item(0) != null)
102                     address = resultNodeList.item(0).getTextContent() + " ";
103
104                 // Get Route Of Location - combine with street number to get address
105                 resultNodeList = (NodeList) xpath.evaluate("
/GeocodeResponse/result[1]/address_component[type/text()='route']/long_name", geocoderResultDocument, XPathConstants.NO
106                 if (resultNodeList.item(0) != null)
107                 {
108                     address += resultNodeList.item(0).getTextContent();
109                     location.setAddress(address);
110                 }
111
112                 // Get City Of Location

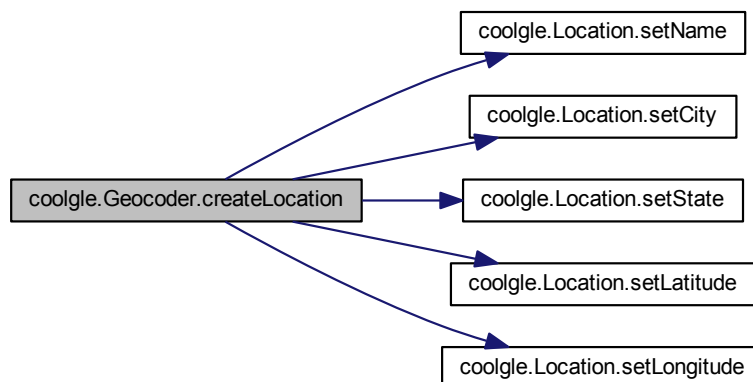
```

```

113         resultNodeList = (NodeList) xpath.evaluate("
/GeocodeResponse/result[1]/address_component[type/text()='locality']/long_name", geocoderResultDocument, XPathConstants.NODESET);
114         if (resultNodeList.item(0) != null)
115             location.setCity(resultNodeList.item(0).getTextContent());
116
117         // Get State Of Location
118         resultNodeList = (NodeList) xpath.evaluate("
/GeocodeResponse/result[1]/address_component[type/text()='administrative_area_level_1']/long_name", geocoderResultDocument, XPathConstants.NODESET);
119         if (resultNodeList.item(0) != null)
120             location.setState(resultNodeList.item(0).getTextContent());
121
122         // Get Lat/Long
123         resultNodeList = (NodeList) xpath.evaluate("
/GeocodeResponse/result[1]/geometry/location/lat", geocoderResultDocument, XPathConstants.NODESET);
124         if (resultNodeList.item(0) != null)
125             location.setLatitude(Double.parseDouble(resultNodeList.item(0).getTextContent()));
126
127         resultNodeList = (NodeList) xpath.evaluate("
/GeocodeResponse/result[1]/geometry/location/lng", geocoderResultDocument, XPathConstants.NODESET);
128         if (resultNodeList.item(0) != null)
129             location.setLongitude(Double.parseDouble(resultNodeList.item(0).getTextContent()));
130     } // End else
131 } // End try
132 catch (IOException e)
133 {
134     JOptionPane.showMessageDialog(null, "Geocoder - IOException", "Failure",
JOptionPane.ERROR_MESSAGE);
135 }
136 catch (XPathExpressionException e)
137 {
138     JOptionPane.showMessageDialog(null, "Geocoder - XPathExpressionException", "Failure",
JOptionPane.ERROR_MESSAGE);
139 }
140 catch (ParserConfigurationException e)
141 {
142     JOptionPane.showMessageDialog(null, "Geocoder - ParserConfigurationException", "Failure",
JOptionPane.ERROR_MESSAGE);
143 }
144 catch (SAXException e)
145 {
146     JOptionPane.showMessageDialog(null, "Geocoder - SAXException", "Failure",
JOptionPane.ERROR_MESSAGE);
147 }
148 return location;
149 } // end of createLocation method

```

Here is the call graph for this function:



The documentation for this class was generated from the following file:

- [src/coolgle/Geocoder.java](#)

## 6.8 coolgle.KmlCreator Class Reference

### Public Member Functions

- [KmlCreator](#) (String file)
- File [FillKml](#) ([Search](#) search)
- void [openKml](#) (File kmlFile)
- String [printTime](#) (double timeToPrint)

### 6.8.1 Constructor & Destructor Documentation

#### 6.8.1.1 coolgle.KmlCreator.KmlCreator ( String file )

[KmlCreator](#) Description - constructor for kmlCreator Parameter - name you want the file to be called. Ex : "newKM-L.kml"

```
54     {
55         fileName = file;
56     }
```

### 6.8.2 Member Function Documentation

#### 6.8.2.1 File coolgle.KmlCreator.FillKml ( Search search )

[FillKml](#) Description - Creates a path in the kml file with points given in input search object. Parameters - [Search](#) object Return - Kml File, of type File. Must import java.io.File to use it outside this class. Pre - ArrayList in search has at least 1 mid location, a start location, and a end location All locations in list have valid latitude and longitude. Note : [Search](#) should be optimized.

Post -kml file is ready to be sent into the openKml function.

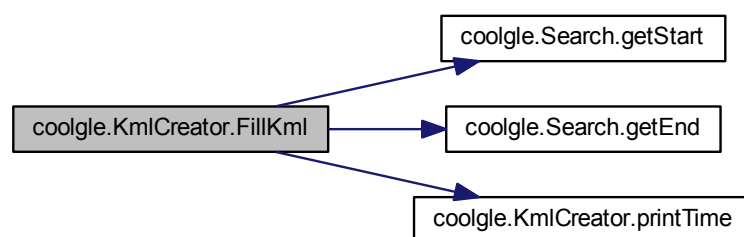
```
71     {
72         // Grab information from search
73         ArrayList<Location> locations = search.getMidLocations();
74
75         // Create The KML File
76         File newFile = new File(fileName);
77         try
78         {
79             // Create the kml object and document object to edit it
80             final Kml kml = new Kml();
81             final Document document = new Document();
82             kml.setFeature(document);
83
84             // Set the FileName of the document
85             document.setName(fileName);
86             document.setOpen(true);
87
88             // Create a new Path
89             final Placemark placemark1 = new Placemark();
90             document.getFeature().add(placemark1);
91
92             // Naming
93             placemark1.setName("Shortest Path");
94             final LineString linestring1 = new LineString();
95             placemark1.setGeometry(linestring1);
96             linestring1.setExtrude(false);
97             linestring1.setTessellate(true);
98
99             List<Coordinate> coord1 = new ArrayList<Coordinate>();
100
101             // Add Start location to path
102             coord1.add(new Coordinate(search.getStart().getLongitude(), search.getStart().getLatitude()));
103
104             // Add each Mid location to the path
105             for (int i = 0; i < locations.size(); i++)
106             {
107                 coord1.add(new Coordinate(locations.get(i).getLongitude(), locations.get(i).getLatitude()));
108             }
109 }
```

```

110         // Add End location to path
111         coordl.add(new Coordinate(search.getEnd().getLongitude(),search.getEnd().getLatitude()));
112
113         linestring1.setCoordinates(coordl);
114
115         // Add pointers to each location to display more information about them.
116         double time = search.getStartTime();
117         int distance = 0;
118
119         // Points for the Locations
120
121         // For this, we will add the starting and ending locations to our arraylist of mid locations.
122         locations.add(0,search.getStart());
123         locations.add(locations.size(),search.getEnd());
124         int i = 0;
125         for ( i = 0; i < locations.size(); i++ )
126         {
127             double newDist = 0;
128             if (i > 0)
129                 newDist = locations.get(i).distanceTo(locations.get(i-1));
130             distance += Math.round(newDist);
131             if ( newDist != 0 )
132                 time += (newDist / SPEED);
133
134             document.createAndAddStyle()
135                 .createAndSetBalloonStyle();
136
137             Placemark placemark = document.createAndAddPlacemark()
138                 .withName(locations.get(i).getName() + " (" + i + ")") //This is the name that shows when
you're not clicked in on it.
139                 .withDescription("Estimated Arrival Time : <br> " + printTime(time)
140                     + "<br>Distance Traveled : <br> " + distance + " miles");
141
142             Point point = placemark.createAndSetPoint();
143             List<Coordinate> coord = point.createAndSetCoordinates();
144             coord.add( coordl.get(i) );
145         }
146
147         // Copy the data above into the file
148         kml.marshal(newFile);
149     }
150     catch (FileNotFoundException ex)
151     {
152         JOptionPane.showMessageDialog(null, "KML - FileNotFoundException", "Failure",
JOptionPane.ERROR_MESSAGE);
153     }
154 }
155
156     return newFile;
157 }

```

Here is the call graph for this function:



### 6.8.2.2 void coolgle.KmlCreator.openKml ( File *kmlFile* )

**openKml Description** - Opens the kml file in Google earth  
**Parameters** - File to be opened, should be the same as the file returned by FillKml function.  
**Pre** - Google earth is installed on end users computer.  
**Post** - Google Earth will be opened, centered in the path.

```

168     {
169         // Open the File with it's default program
170         if (Desktop.isDesktopSupported())
171         {
172             try
173             {
174                 Desktop.getDesktop().open(kmlFile);
175             }
176             catch (IOException ex)
177             {
178                 JOptionPane.showMessageDialog(null, "KML - IOException", "Failure",
179                 JOptionPane.ERROR_MESSAGE);
180             }
181         }
182         else
183         {
184             // If error opening file, post error message to console.
185             JOptionPane.showMessageDialog(null, "Cannot open KML file", "Failure",
186             JOptionPane.ERROR_MESSAGE);
187         }
188     }

```

### 6.8.2.3 String coolgle.KmlCreator.printTime ( double *timeToPrint* )

**printTime** Description - Gives a nicely formatted time string Parameters - int time from search, for example, 8.50 ( = 8:30 AM) Return - string msg for example : 8:30 AM Pre - Google earth is installed on end users computer. Post -Google Earth will be opened, centered in the path.

```

198     {
199         String msgTime = "";
200
201         String AMPM = "AM";
202
203         // If time is over a day, return to miliary time again
204         if (timeToPrint >= 24)
205             timeToPrint = timeToPrint % 24;
206         // Get Hours :
207         double hours = Math.floor(timeToPrint);
208         if (hours > 12)
209         {
210             AMPM = "PM";
211             hours -= 12;
212             timeToPrint -= 12;
213         }
214         if (hours == 12)
215             AMPM = "PM";
216         if (hours == 12)
217         {
218             AMPM = "AM";
219             hours = 12;
220         }
221
222         DecimalFormat df = new DecimalFormat("00");
223
224         // Get Minutes :
225         timeToPrint -= hours; // Just get minute part
226         double mins = timeToPrint;
227         mins *= 60;
228         DecimalFormat df2 = new DecimalFormat("00");
229
230         msgTime = df.format(hours) + ":" + df2.format(mins) + " " + AMPM;
231
232         return msgTime;
233     }

```

Here is the caller graph for this function:





The documentation for this class was generated from the following file:

- [src/coolgle/KmlCreator.java](#)

## 6.9 coolgle.Location Class Reference

### Public Member Functions

- [Location](#) ()
- [Location](#) ([Location](#) loc)
- [Location](#) (String fileInput)
- [Location](#) (String name, double latitude, double longitude, String city, String state, String address)
- [Location](#) (String name, String address, String city, String state, [Coord](#) coords)
- String [getName](#) ()
- void [setName](#) (String name)
- [Coord](#) [getCoord](#) ()
- void [setCoord](#) ([Coord](#) coord)
- String [getAddress](#) ()
- void [setAddress](#) (String address)
- String [getCity](#) ()
- void [setCity](#) (String city)
- String [getState](#) ()
- void [setState](#) (String state)
- double [getLatitude](#) ()
- double [getLongitude](#) ()
- void [setLongitude](#) (double newLongitude)
- void [setLatitude](#) (double newLatitude)
- String [toString](#) ()
- String [fileToString](#) ()
- boolean [printToFile](#) (String fileName, boolean allowRepeats)
- double [distanceTo](#) ([Location](#) otherLocation)
- boolean [isSame](#) ([Location](#) otherLocation)
- boolean [printToFile](#) (String fileName)

### 6.9.1 Constructor & Destructor Documentation

#### 6.9.1.1 coolgle.Location.Location ( )

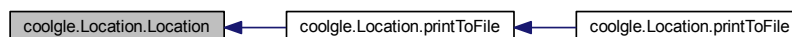
Default Constructor for location object :

```

39     {
40         this("", "", "", "", new Coord(0,0));
41     }

```

Here is the caller graph for this function:



### 6.9.1.2 coolgle.Location.Location ( Location loc )

Clone Constructor for location object :

```

46     {
47         this(loc.name, loc.city, loc.state, loc.address, loc.coord);
48     }

```

### 6.9.1.3 coolgle.Location.Location ( String fileInput )

Constructor - needs valid non null string; Description parses input string

```

55     {
56         String[] data = fileInput.split("\\|");
57
58         name = data[0];
59         address = data[1];
60         city = data[2];
61         state = data[3];
62         coord = new Coord((Double.parseDouble(data[4])), (Double.parseDouble(data[5])) );
63     }

```

### 6.9.1.4 coolgle.Location.Location ( String name, double latitude, double longitude, String city, String state, String address )

Constructor - needs valid non null entries Description creates search with param values(Coord created in constructor)

```

71     {
72         this(name, address, city, state, new Coord(latitude, longitude));
73     }

```

### 6.9.1.5 coolgle.Location.Location ( String name, String address, String city, String state, Coord coords )

Constructor - needs valid non null entries Description creates search with param values(Coord created out constructor)

```

80     {
81         this.name = name;
82         coord = new Coord(coords.getLatitude(), coords.getLongitude());
83         this.city = city;
84         this.state = state;
85         this.address = address;
86     }

```

Here is the call graph for this function:



## 6.9.2 Member Function Documentation

### 6.9.2.1 double coolgle.Location.distanceTo ( Location *otherLocation* )

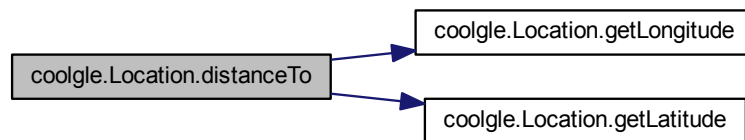
Description- Returns the distance in miles between the two locations

```

264     {
265         double distance = 0.0;
266         double lon1 = getLocation();
267         double lat1 = getLatitude();
268         double lon2 = otherLocation.getLongitude();
269         double lat2 = otherLocation.getLatitude();
270
271         double theta = lon1 - lon2;
272         distance = Math.sin(convertToRadians(lat1)) * Math.sin(convertToRadians(lat2)) +
273             Math.cos(convertToRadians(lat1)) * Math.cos(convertToRadians(lat2)) *
274             Math.cos(convertToRadians(theta));
275         distance = Math.acos(distance);
276         distance = convertToDegrees(distance);
277         distance = distance * 60 * 1.1515;
278
279         return distance;
280     }

```

Here is the call graph for this function:



### 6.9.2.2 String coolgle.Location.fileToString ( )

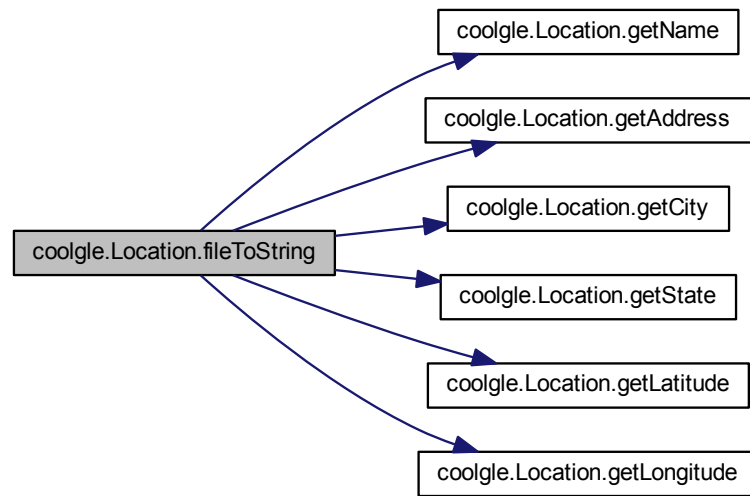
Description- toString for printing to database file

```

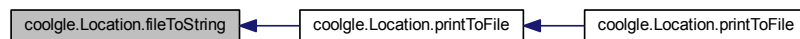
199     {
200         String returnString = "";
201         returnString += getName() + " | ";
202         returnString += getAddress() + " | ";
203         returnString += getCity() + " | ";
204         returnString += getState() + " | ";
205         returnString += getLatitude() + " | ";
206         returnString += getLongitude();
207         return returnString;
208     }

```

Here is the call graph for this function:



Here is the caller graph for this function:



### 6.9.2.3 String coolgle.Location.getAddress ( )

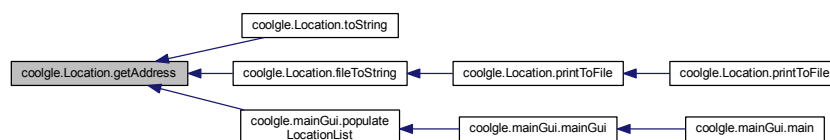
#### Returns

the address

```

116         {
117             return address;
118         }
  
```

Here is the caller graph for this function:



## 6.9.2.4 String coolgle.Location.getCity ( )

## Returns

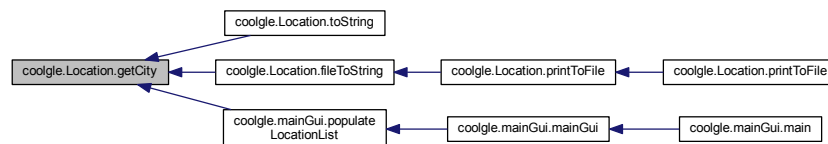
the city

```

128         {
129             return city;
130         }

```

Here is the caller graph for this function:



## 6.9.2.5 Coord coolgle.Location.getCoord ( )

## Returns

the coord

```

104         {
105             return coord;
106         }

```

## 6.9.2.6 double coolgle.Location.getLatitude ( )

## Parameters

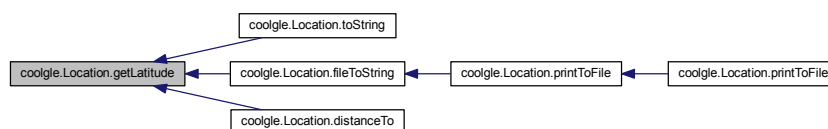
<i>get</i>	latitude
------------	----------

```

153     {
154         return coord.getLatitude();
155     }

```

Here is the caller graph for this function:



## 6.9.2.7 double coolgle.Location.getLongitude ( )

## Parameters

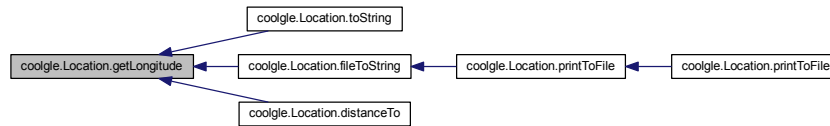
<i>get</i>	longitude
------------	-----------

```

160     {
161         return coord.getLongitude();
162     }

```

Here is the caller graph for this function:



## 6.9.2.8 String coolgle.Location.getName ( )

## Returns

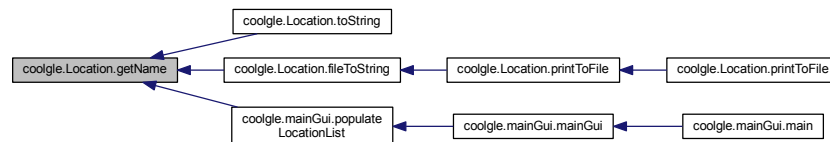
the name

```

91     {
92         return name;
93     }

```

Here is the caller graph for this function:



## 6.9.2.9 String coolgle.Location.getState ( )

## Returns

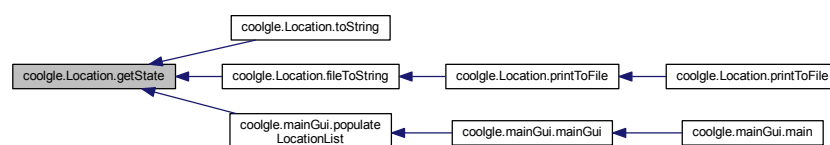
the state

```

140     {
141         return state;
142     }

```

Here is the caller graph for this function:



### 6.9.2.10 boolean coolgle.Location.isSame ( Location *otherLocation* )

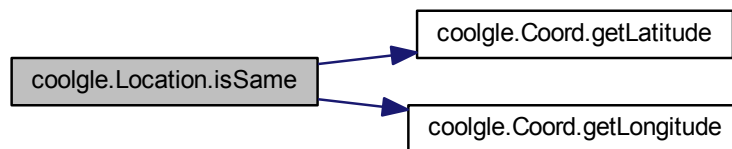
Description- checks for same location within a search

```

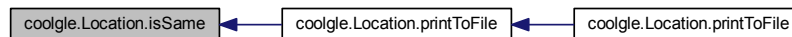
300  {
301      boolean same = true;
302
303      //We only check the lat and long in case they changed the name
304      if ( coord.getLatitude() != otherLocation.getLatitude() )
305          same = false;
306      if ( coord.getLongitude() != otherLocation.getLongitude() )
307          same = false;
308
309      return same;
310  }

```

Here is the call graph for this function:



Here is the caller graph for this function:



### 6.9.2.11 boolean coolgle.Location.printToFile ( String *fileName*, boolean *allowRepeats* )

Description- Returns true if no errors, false if otherwise

```

215  {
216      boolean success = true;
217
218      boolean alreadyInFile = false;
219      try
220      {
221          // Check the file to see if the location already exists :
222          BufferedReader br;
223
224          br = new BufferedReader(new FileReader("LocationDatabase.txt"));
225          String locationString;
226          while ((locationString = br.readLine()) != null)
227          {
228              Location newLocation = new Location(locationString);
229              if (isSame(newLocation)) // The location already exists in the file.
230              {
231                  alreadyInFile = true;
232              }
233          }
234
235          if (!alreadyInFile || (alreadyInFile && allowRepeats))
236          {

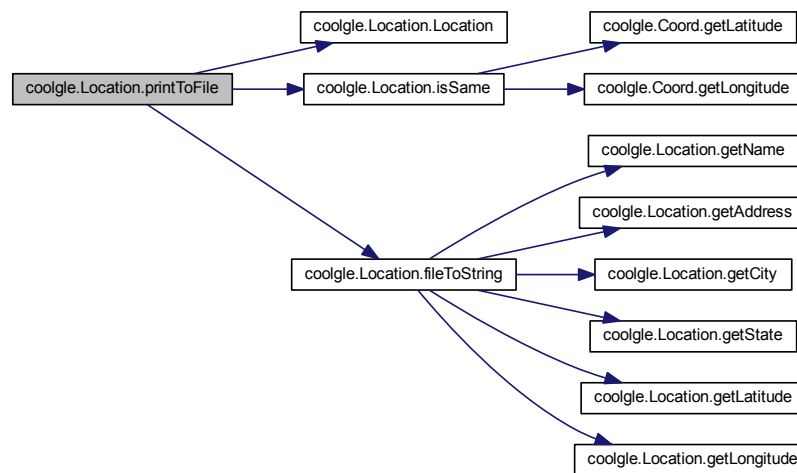
```

```

237         BufferedWriter writer = new BufferedWriter(new FileWriter(fileName, true));
238         writer.append(fileToString());
239         writer.newLine();
240         writer.close();
241     }
242     else // Already in file, don't add it again, show warning.
243     {
244         JOptionPane.showMessageDialog(null, "This location is already in the file"
245             , "Failure", JOptionPane.ERROR_MESSAGE);
246         success = false;
247     }
248     br.close();
249 }
250 catch (FileNotFoundException e)
251 {
252     JOptionPane.showMessageDialog(null, "Database file not found", "Failure",
253         JOptionPane.ERROR_MESSAGE);
254 }
255 catch (IOException e)
256 {
257     success = false;
258 }
259 return success;

```

Here is the call graph for this function:



Here is the caller graph for this function:



### 6.9.2.12 boolean coolgle.Location.printToFile ( String fileName )

Description- repopulates filesystem from admingui

```

315 {

```

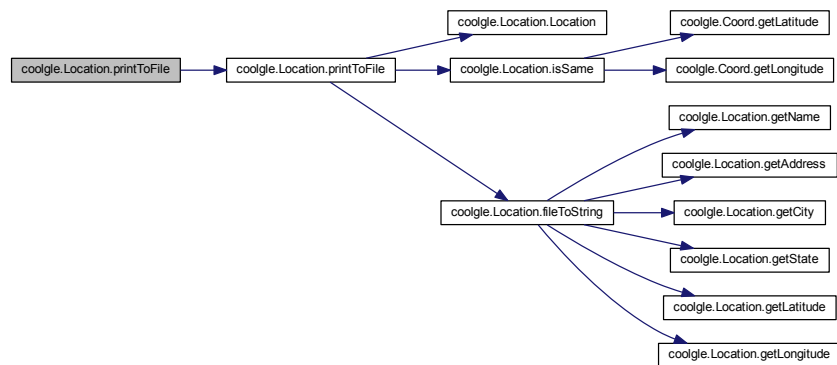


```

316         return printToFile(fileName, true);
317     }

```

Here is the call graph for this function:



#### 6.9.2.13 void coolgle.Location.setAddress ( String address )

##### Parameters

<i>address</i>	the address to set
----------------	--------------------

```

122     {
123         this.address = address;
124     }

```

#### 6.9.2.14 void coolgle.Location.setCity ( String city )

##### Parameters

<i>city</i>	the city to set
-------------	-----------------

```

134     {
135         this.city = city;
136     }

```

Here is the caller graph for this function:



#### 6.9.2.15 void coolgle.Location.setCoord ( Coord coord )

## Parameters

<i>coord</i>	the coord to set
--------------	------------------

```
110         {  
111             this.coord = coord;  
112         }
```

6.9.2.16 void coolgle.Location.setLatitude ( double *newLatitude* )

## Parameters

<i>state</i>	the latitude to set
--------------	---------------------

```
175     {  
176         coord.setLatitude(newLatitude);  
177     }
```

Here is the caller graph for this function:

6.9.2.17 void coolgle.Location.setLongitude ( double *newLongitude* )

## Parameters

<i>state</i>	the longitude to set
--------------	----------------------

```
168     {  
169         coord.setLongitude(newLongitude);  
170     }
```

Here is the caller graph for this function:

6.9.2.18 void coolgle.Location.setName ( String *name* )

## Parameters

<i>name</i>	the name to set
-------------	-----------------

```

97         {
98             this.name = name;
99         }

```

Here is the caller graph for this function:



## 6.9.2.19 void coolgle.Location.setState ( String state )

## Parameters

<i>state</i>	the state to set
--------------	------------------

```

146         {
147             this.state = state;
148         }

```

Here is the caller graph for this function:



## 6.9.2.20 String coolgle.Location.toString ( )

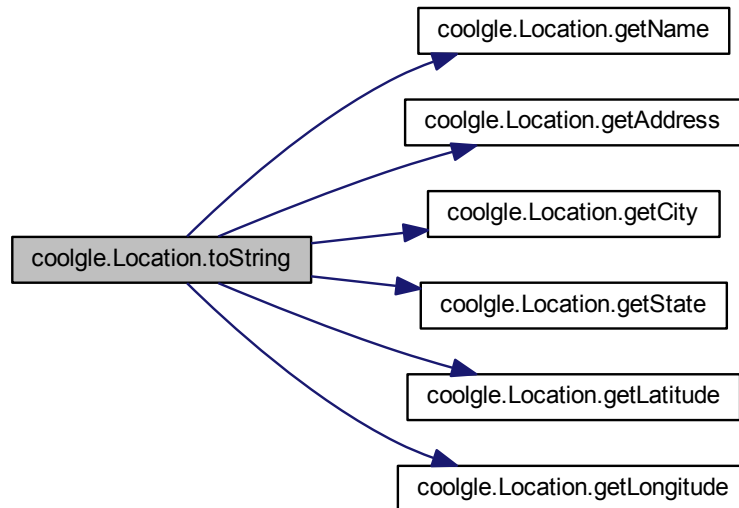
Description- Override of toString

```

184     {
185         String returnString = "";
186         returnString += "Name : " + getName() + "\n";
187         returnString += "Address : " + getAddress() + "\n";
188         returnString += "City : " + getCity() + "\n";
189         returnString += "State : " + getState() + "\n";
190         returnString += "Latitude : " + getLatitude() + "\n";
191         returnString += "Longitude : " + getLongitude() + "\n";
192         return returnString;
193     }

```

Here is the call graph for this function:

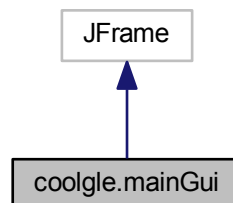


The documentation for this class was generated from the following file:

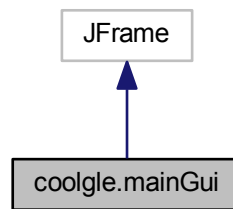
- [src/coolgle/Location.java](#)

## 6.10 coolgle.mainGui Class Reference

Inheritance diagram for `coolgle.mainGui`:



Collaboration diagram for coolgle.mainGui:



## Public Member Functions

- `mainGui` (String userName)
- void `logOut` ()
- void `changeTrip` ()
- void `updateStartingTime` ()
- void `updateStart` ()
- void `updateMid` ()
- void `removeSelectedMidTrip` ()
- void `updateEnd` ()
- void `populateLocationList` (int filterCategory, String FilterText)
- void `validateTrip` ()
- void `prepareTrip` ()
- void `viewInfomation` ()
- void `updateVisualDisplay` ()
- void `clearSearch` ()
- void `viewHelp` ()

## Static Public Member Functions

- static void `main` (String args[])

## 6.10.1 Constructor & Destructor Documentation

### 6.10.1.1 coolgle.mainGui.mainGui ( String userName )

Constructor Creates new form `mainGui`

```

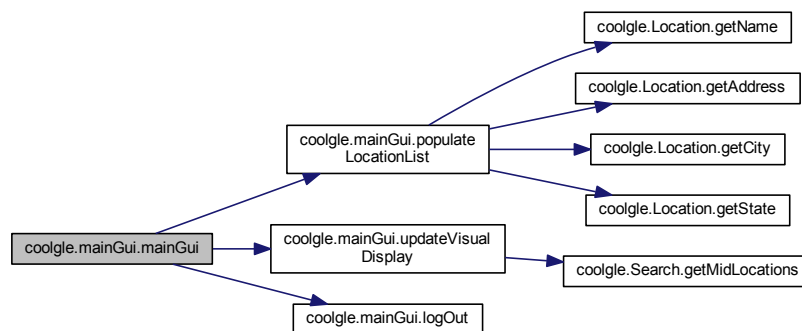
83     {
84         Dimension screen = Toolkit.getDefaultToolkit().getScreenSize();
85         Dimension frm = super.getSize();
86         int xpos = (int) (screen.getWidth() / 8 - frm.getWidth() / 2);
87         int ypos = (int) (screen.getHeight() / 30);
88         super.setLocation(xpos, ypos);
89         initComponents();
90
91         // Populate Location List in bottom left
92         populateLocationList(filterCategory.getSelectedIndex(), filterText.getText());
93
94         // Populate previous searches. TODO
95
96         // Create the search object, to be edited and sent to KML Creator
  
```

```

97     search = new Search(userName);
98     user = userName;
99
100    updateVisualDisplay();
101
102
103    //Allows the logout or close popup to appear when closing window
104    this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
105    this.addWindowListener(new java.awt.event.WindowAdapter()
106    {
107    @Override
108    public void windowClosing(java.awt.event.WindowEvent windowEvent)
109    {
110        Object[] options = {"Log out","Exit"};
111        int n = JOptionPane.showOptionDialog(null,
112        "Exit?", "Exit",JOptionPane.YES_NO_CANCEL_OPTION,
113        JOptionPane.QUESTION_MESSAGE,
114        null,
115        options,
116        options[1]);
117
118        if (n == 1)
119        {
120            System.exit(0);
121        }
122        else // Log out
123        {
124            logOut();
125        }
126    }
127    });

```

Here is the call graph for this function:



Here is the caller graph for this function:



## 6.10.2 Member Function Documentation

### 6.10.2.1 void coolgle.mainGui.changeTrip ( )

Description- When something is selected from pervious search dropdown :

## Parameters

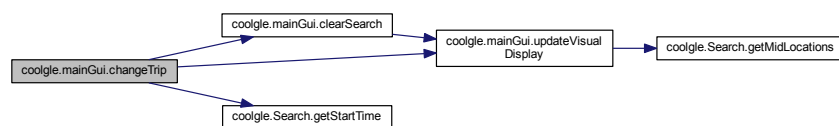
<i>evt</i>	
------------	--

```

730     {
731         FileManager fm = new FileManager();
732         if ( prevList.getSelectedIndex() > 0 )
733         {
734             search = fm.getPrev( user, prevList.getSelectedIndex() );
735             search.setOptimized(false);
736             user = search.getUserName();
737         }
738         if ( prevList.getSelectedIndex() == 0 )
739         {
740             clearSearch();
741         }
742         updateVisualDisplay();
743
744         // Select correct time from time box
745         times.setSelectedIndex((int)search.getStartTime() - 8);
746     }

```

Here is the call graph for this function:



## 6.10.2.2 void coolgle.mainGui.clearSearch ( )

Description- to be called when the user presses the clear search button,

clears fields of search & updates the visuals in the right pane.

```

1034     {
1035         search.clearData();
1036
1037         // Set time dropdown back to 8AM
1038         times.setSelectedIndex(0);
1039
1040         midLocationsList.removeAll();
1041
1042         updateVisualDisplay();
1043     }

```

Here is the call graph for this function:



Here is the caller graph for this function:



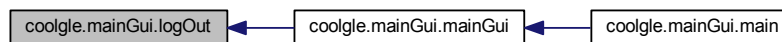
#### 6.10.2.3 void coolgle.mainGui.logOut ( )

logOut Description - To be called when user exits screen, asks if user wants to log out or completely close

```

710     {
711         SignInGui logOut = new SignInGui();
712         logOut.setVisible(true);
713         this.setVisible(false);
714     }
  
```

Here is the caller graph for this function:



#### 6.10.2.4 static void coolgle.mainGui.main ( String args[] ) [static]

##### MAIN TESTER METHOD

##### Parameters

<i>args</i>	the command line arguments
-------------	----------------------------

```

1062     {
1063         /* Set the Nimbus look and feel */
1064         //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
1065         /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
1066          * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
1067          */
1068         try {
1069             for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.
getInstalledLookAndFeels()) {
1070                 if ("Nimbus".equals(info.getName())) {
1071                     javax.swing.UIManager.setLookAndFeel(info.getClassName());
1072                     break;
1073                 }
1074             }
1075             } catch (ClassNotFoundException ex) {
1076                 java.util.logging.Logger.getLogger(mainGui.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
1077             } catch (InstantiationException ex) {
1078                 java.util.logging.Logger.getLogger(mainGui.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
1079             } catch (IllegalAccessException ex) {
1080                 java.util.logging.Logger.getLogger(mainGui.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
1081             } catch (javax.swing.UnsupportedLookAndFeelException ex) {
1082                 java.util.logging.Logger.getLogger(mainGui.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
  
```

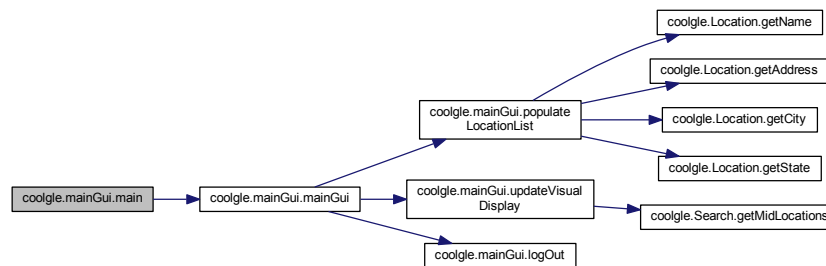


```

1083     }
1084     //</editor-fold>
1085
1086     /* Create and display the form */
1087     java.awt.EventQueue.invokeLater(new Runnable()
1088     {
1089         public void run()
1090         {
1091             new mainGui(user).setVisible(true);
1092         }
1093     });
1094 }

```

Here is the call graph for this function:



#### 6.10.2.5 void coolgle.mainGui.populateLocationList ( int filterCategory, String FilterText )

Description populateLocationList Description - Populates the location list (the list of locations the user can choose to add to their current trip). Takes in a filter category (explained above) and a filterText which is the text of the filterText field. This is called when the screen is first loaded, and every time the user makes a change to the filterText field  
additional:filterCategory : 0 - Name 1 - Address 2 - City 3 - State

```

845     {
846         try
847         {
848             // Clear out current list
849             locationList.clear();
850
851             // Fill in Location List
852             DefaultListModel listModel;
853             listModel = new DefaultListModel();
854             MainLocationDisplay.setModel(listModel);
855             BufferedReader br;
856             try
857             {
858                 br = new BufferedReader(new FileReader("LocationDatabase.txt"));
859                 String locationString;
860                 //ArrayList<Location> DatabaseLocations = new ArrayList<Location>();
861                 // Read file Location by location
862                 while ((locationString = br.readLine()) != null)
863                 {
864                     Location newLocation = new Location(locationString);
865                     // Does it match the filter? :
866                     boolean toAdd = false;
867                     if ( filterCategory == 0 && newLocation.getName().toLowerCase().contains(FilterText.
868 toLowerCase()) )
869                     {
870                         toAdd = true;
871                     }
872                     if ( filterCategory == 1 && newLocation.getAddress().toLowerCase().contains(FilterText.
873 toLowerCase()) )
874                     {
875                         toAdd = true;
876                     }
877                     if ( filterCategory == 2 && newLocation.getCity().toLowerCase().contains(FilterText.
878 toLowerCase()) )
879                     {
880                         toAdd = true;
881                     }
882                 }
883             }
884             catch (IOException e)
885             {
886                 // Handle exception
887             }
888         }
889         catch (Exception e)
890         {
891             // Handle exception
892         }
893     }
894 }

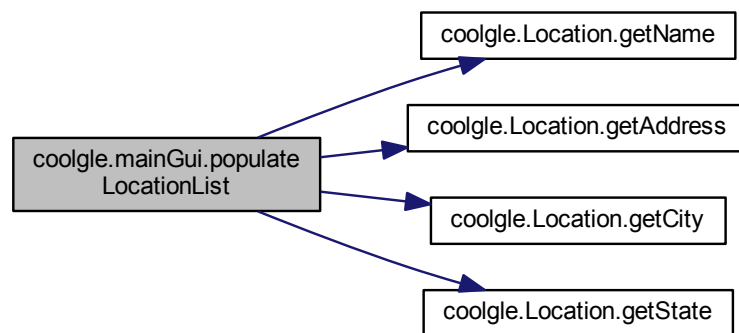
```

```

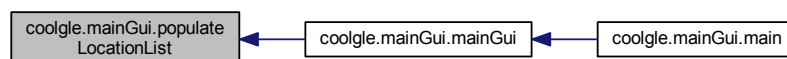
879         if ( filterCategory == 3 && newLocation.getState().toLowerCase().contains(FilterText.
toLowerCase()) )
880         {
881             toAdd = true;
882         }
883         if (toAdd)
884         {
885             locationList.add(newLocation);
886             listModel.addElement(newLocation.getName());
887         }
888     }
889     br.close();
890 }
891 catch (FileNotFoundException ex)
892 {
893     JOptionPane.showMessageDialog(null, "Main - FileNotFound", "Failure",
JOptionPane.ERROR_MESSAGE);
894 }
895 }
896 catch (IOException ex)
897 {
898     JOptionPane.showMessageDialog(null, "Main - IOException", "Failure", JOptionPane.ERROR_MESSAGE)
;
899 }
900 }

```

Here is the call graph for this function:



Here is the caller graph for this function:



#### 6.10.2.6 void coolgle.mainGui.prepareTrip ( )

Description-Called after trip is validated from the `validateTrip()` function.

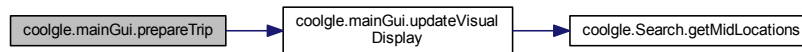
Creates a [SearchManager](#) object to sort the trip and then passes the search to the `kmlCreator` class which creates a kml file based on the search and opens the file.

```

942     {
943         SearchManager newSearchManager = new SearchManager();
944         newSearchManager.sortShortestDistance(search);
945
946         \n"
947         + "Would you like to save it and view it on google earth now?");
948         if (confirm == 0) // Yes
949         {
950             // Save this search in previous searches.
951             FileManager fm = new FileManager();
952             fm.addSearch(search);
953
954             updateVisualDisplay();
955
956             // Create and open KML file
957             KmlCreator newKml = new KmlCreator("Search1.kml");
958             newKml.openKml(newKml.FillKml(search));
959         }
960     }

```

Here is the call graph for this function:



Here is the caller graph for this function:



#### 6.10.2.7 void coolgle.mainGui.removeSelectedMidTrip ( )

Description- Removes the selected item from the right list from the current trip.

```

800     {
801         //If something is selected, remove it
802         if (midLocationsList.getSelectedIndex() != -1)
803         {
804             search.removeAtPosition( midLocationsList.getSelectedIndex() );
805             search.setOptimized(false);
806         }
807         else
808         {
809             JOptionPane.showMessageDialog(null, "Nothing is selected", "Failure", JOptionPane.ERROR_MESSAGE);
810         }
811         updateVisualDisplay();
812     }

```

Here is the call graph for this function:



#### 6.10.2.8 void coolgle.mainGui.updateEnd ( )

Description- Sets the searches starting location to the currently selected one

```

818     {
819         //Make sure location isn't already in search.
820         if ( MainLocationDisplay.getSelectedIndex() != -1 && !search.isInSearch(locationList.get (
MainLocationDisplay.getSelectedIndex())) )
821         {
822             search.setEnd(locationList.get (MainLocationDisplay.getSelectedIndex()));
823             search.setOptimized(false);
824             updateVisualDisplay();
825         }
826         else //Already in search
827             JOptionPane.showMessageDialog(null, "Location already in trip", "Failure",
JOptionPane.ERROR_MESSAGE);
828     }
829 
```

Here is the call graph for this function:



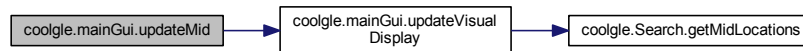
#### 6.10.2.9 void coolgle.mainGui.updateMid ( )

Description- Adds the selected item in the left list to the current search, updates the right list. Shows error if the location is already in the trip.

```

782     {
783         //Make sure location isn't already in search.
784         if ( MainLocationDisplay.getSelectedIndex() != -1 && !search.isInSearch(locationList.get (
MainLocationDisplay.getSelectedIndex())) )
785         {
786             search.addLocation(locationList.get (MainLocationDisplay.getSelectedIndex()));
787             search.setOptimized(false);
788             updateVisualDisplay();
789         }
790         else //Already in search
791         {
792             JOptionPane.showMessageDialog(null, "Location already in trip", "Failure",
JOptionPane.ERROR_MESSAGE);
793         }
794     }
  
```

Here is the call graph for this function:



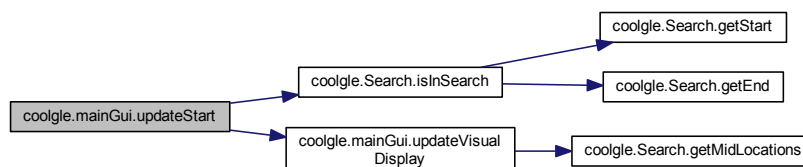
#### 6.10.2.10 void coolgle.mainGui.updateStart ( )

Description- Sets the searches starting location to the currently selected one

```

763     {
764         //Make sure location isn't already in search.
765         if ( !search.isInSearch(locationList.get(MainLocationDisplay.getSelectedIndex())) )
766         {
767             search.setStart(locationList.get(MainLocationDisplay.getSelectedIndex()));
768             search.setOptimized(false);
769             updateVisualDisplay();
770         }
771         else //Already in search
772             JOptionPane.showMessageDialog(null, "Location already in trip", "Failure",
773                                         JOptionPane.ERROR_MESSAGE);
774     }
  
```

Here is the call graph for this function:



#### 6.10.2.11 void coolgle.mainGui.updateStartingTime ( )

Description- Called when user selects a time in the starting time drop-down Sets the starting time member vairable in the search and updates the right pane.

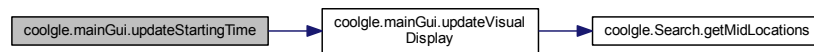
Parameters

<i>evt</i>	
------------	--

```

754     {
755         search.setStartTime(8 + times.getSelectedIndex());
756         updateVisualDisplay();
757     }
  
```

Here is the call graph for this function:



#### 6.10.2.12 void coolgle.mainGui.updateVisualDisplay ( )

Description- Called after any variable in the search object is changed.

It will update the right pane of the main gui to account for the change in information.

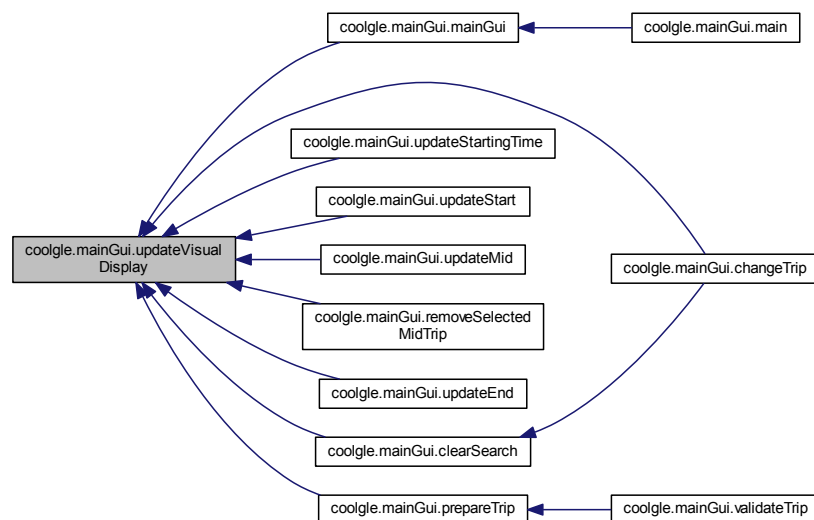
```

982     {
983         // Set Time
984         timeDisplay.setText( times.getSelectedItem().toString() );
985
986         // Set Start Location
987         startLocationDisplay.setText( search.getStart().getName() );
988
989         // Set End Location
990         endLocationDisplay.setText( search.getEnd().getName() );
991
992         DefaultListModel listModel;
993         listModel = new DefaultListModel();
994         midLocationsList.setModel( listModel );
995         for (int i = 0; i < search.getNumMidLocations(); i++)
996             listModel.addElement( search.getMidLocations().get(i).getName() );
997
998         // Update Previous Search Dropdown :
999         FileManager fml = new FileManager();
1000
1001
1002         // Save the current selection of the previous search dropdown :
1003         //int oldSelection = prevList.getSelectedIndex();
1004
1005         DefaultComboBoxModel comboModel;
1006
1007         comboModel = new DefaultComboBoxModel();
1008         comboModel.addElement( "Custom Search" );
1009
1010         Search pastSearch = new Search();
1011         int i = 1;
1012         boolean cont = true;
1013         while ( cont )
1014         {
1015             pastSearch = fml.getPrev( user, i );
1016             if ( pastSearch.getNumMidLocations() != 0 )
1017             {
1018                 comboModel.addElement( pastSearch.toString() );
1019                 i++;
1020             }
1021             else
1022             {
1023                 cont = false;
1024             }
1025         }
1026         prevList.setModel( comboModel );
1027
1028         //prevList.setSelectedIndex( oldSelection );
1029     }
  
```

Here is the call graph for this function:



Here is the caller graph for this function:



#### 6.10.2.13 void coolgle.mainGui.validateTrip ( )

Description - Called when user clicks map it button, verifies that the trip has a starting/ending location and at least one mid location. If so, calls `prepareTrip` which sorts the trip & sends it to the kml creator.

```

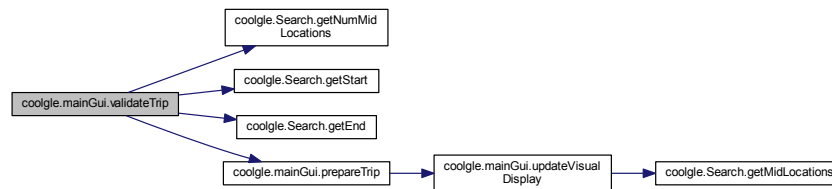
909     {
910         //Validate trip
911         boolean validTrip = true;
912         if (search.getNumMidLocations() <= 0)
913         {
914             JOptionPane.showMessageDialog(null, "Add at least one mid location to continue"
915                 , "Failure", JOptionPane.ERROR_MESSAGE);
916             validTrip = false;
917         }
918         else if (search.getStart().getName().isEmpty() && search.getStart().getAddress().isEmpty())
919         {
920             JOptionPane.showMessageDialog(null, "Invalid Start Location"
921                 , "Failure", JOptionPane.ERROR_MESSAGE);
922             validTrip = false;
923         }
924         else if (search.getEnd().getName().isEmpty() && search.getEnd().getAddress().isEmpty())
925         {
926             JOptionPane.showMessageDialog(null, "Invalid End Location"
927                 , "Failure", JOptionPane.ERROR_MESSAGE);
928             validTrip = false;
929         }
930     }
931     if (validTrip)
  
```

```

932         prepareTrip();
933     }

```

Here is the call graph for this function:



#### 6.10.2.14 void coolgle.mainGui.viewHelp ( )

**viewHelp Description** - Shown when the user presses the question mark icon next to the previous search dropdowns  
Shows a messagebox very briefly explaining the interface.

```

1051     {
1052         JOptionPane.showMessageDialog(null, "Either select a past trip from the dropdown, or \n"
1053             + "use the controls below to plan a new trip.", "Help", JOptionPane.INFORMATION_MESSAGE);
1054     }

```

#### 6.10.2.15 void coolgle.mainGui.viewInformation ( )

**Description**-Called when the user hits the view details button in the left pane of the main gui.

Shows error message if nothing is selected.

```

966     {
967         if ( MainLocationDisplay.getSelectedIndex() != -1 )
968         {
969             AddLocationGui info = new AddLocationGui( locationList.get(MainLocationDisplay.getSelectedIndex
970             ));
971             info.setVisible(true);
972         }
973         else
974         {
975             JOptionPane.showMessageDialog(null, "No location selected."
976                 , "Failure", JOptionPane.ERROR_MESSAGE);
977         }
978     }

```

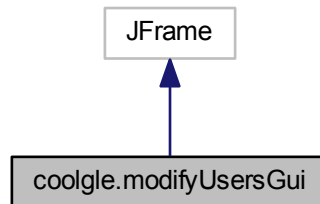
The documentation for this class was generated from the following file:

- [src/coolgle/mainGui.java](#)

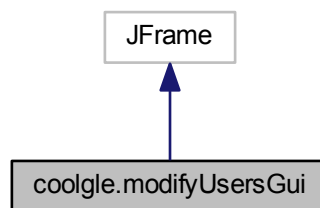


## 6.11 coolgle.modifyUsersGui Class Reference

Inheritance diagram for coolgle.modifyUsersGui:



Collaboration diagram for coolgle.modifyUsersGui:



### Public Member Functions

- `modifyUsersGui` (`adminGui` `mainFrame`, `String` `userName`)

### Static Public Member Functions

- static void `main` (`String` `args`[])

#### 6.11.1 Constructor & Destructor Documentation

##### 6.11.1.1 coolgle.modifyUsersGui.modifyUsersGui ( `adminGui` `mainFrame`, `String` `userName` )

Constructor- Creates new form modifyUsers

```

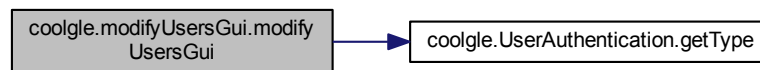
51     {
52         Dimension screen = Toolkit.getDefaultToolkit().getScreenSize();
53         Dimension frm = super.getSize();
54         int xpos = (int) (screen.getWidth() / 8 - frm.getWidth() / 2);
55         int ypos = (int) (screen.getHeight() / 8 - frm.getHeight() / 2);
56         super.setLocation(xpos, ypos);
57         initComponents();
  
```

```

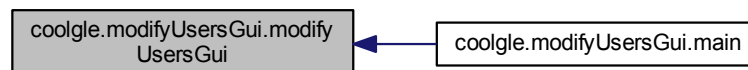
58
59     UserAuthentication getData= new UserAuthentication();
60     pswdText.setText(getData.getPassword(userName));
61     emailText.setText(getData.getEmail(userName));
62     usrText.setText(userName);
63     if (getData.getType(userName) == 1)
64         isAdminCheck.setSelected(true);
65     else
66         isAdminCheck.setSelected(false);
67
68     oldname = userName;
69
70     m_main = mainFrame;
71
72     // Makes it so that you can exit from pop up messages and this window without closing the entire
app
73     this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
74
75     usrText.setEditable(false);
76 }

```

Here is the call graph for this function:



Here is the caller graph for this function:



## 6.11.2 Member Function Documentation

### 6.11.2.1 static void coolgle.modifyUsersGui.main ( String args[] ) [static]

main tester method

```

231 {
232     /* Set the Nimbus look and feel */
233     //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
234     /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
235      * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
236      */
237     try {
238         for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.
getInstalledLookAndFeels()) {
239             if ("Nimbus".equals(info.getName())) {
240                 javax.swing.UIManager.setLookAndFeel(info.getClassName());
241                 break;
242             }
243         }
244     } catch (ClassNotFoundException ex) {
245         java.util.logging.Logger.getLogger(modifyUsersGui.class.getName()).log(
java.util.logging.Level.SEVERE, null, ex);
246     } catch (InstantiationException ex) {
247         java.util.logging.Logger.getLogger(modifyUsersGui.class.getName()).log(

```

```

    java.util.logging.Level.SEVERE, null, ex);
248     } catch (IllegalAccessException ex) {
249         java.util.logging.Logger.getLogger(modifyUsersGui.class.getName()).log(
    java.util.logging.Level.SEVERE, null, ex);
250     } catch (javax.swing.UnsupportedLookAndFeelException ex) {
251         java.util.logging.Logger.getLogger(modifyUsersGui.class.getName()).log(
    java.util.logging.Level.SEVERE, null, ex);
252     }
253     //</editor-fold>
254
255     /* Create and display the form */
256     java.awt.EventQueue.invokeLater(new Runnable()
257     {
258         public void run()
259         {
260             new modifyUsersGui(null, "").setVisible(true);
261         }
262     });
263 }

```

Here is the call graph for this function:

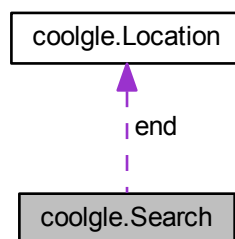


The documentation for this class was generated from the following file:

- [src/coolgle/modifyUsersGui.java](#)

## 6.12 coolgle.Search Class Reference

Collaboration diagram for coolgle.Search:



### Public Member Functions

- [Search \(\)](#)
- [Search \(String un\)](#)
- [Search \(String userName, \[Location\]\(#\) start, \[Location\]\(#\) end, ArrayList< \[Location\]\(#\) > locations\)](#)
- [Search \(String startTime, String userName, \[Location\]\(#\) start, \[Location\]\(#\) end, ArrayList< \[Location\]\(#\) > locations, ArrayList< \[Coord\]\(#\) > shortestPath\)](#)
- boolean [hasShortest \(\)](#)

- void `setOptimized` (boolean isOptimized)
- `ArrayList< Location > getMidLocations` ()
- void `setStartTime` (double newTime)
- double `getStartTime` ()
- `Location` `getStart` ()
- void `setStart` (`Location` newStart)
- `Location` `getEnd` ()
- void `setEnd` (`Location` newEnd)
- String `getUserName` ()
- void `addLocation` (`Location` newLocation)
- boolean `removeLocation` (`Location` removeLocation)
- void `removeAtPosition` (int pos)
- int `getNumMidLocations` ()
- boolean `isInSearch` (`Location` testLocation)
- void `clearData` ()
- String `toString` ()
- String `fileToString` ()
- void `setLocations` (`ArrayList< Location >` locations)
- void `setShortestPath` (`ArrayList< Coord >` shortestPath)
- void `setDistance` (double distance)
- void `setUserName` (String userName)

## 6.12.1 Constructor & Destructor Documentation

### 6.12.1.1 coolgle.Search.Search ( )

constructor for SEARCH object

```

41      {
42          this("", new Location(), new Location(), new ArrayList<Location>());
43      }

```

### 6.12.1.2 coolgle.Search.Search ( String un )

Blank constructor for creating with just userName

```

49      {
50          this(un, new Location(), new Location(), new ArrayList<Location>());
51          //Default Constructor
52      }

```

### 6.12.1.3 coolgle.Search.Search ( String userName, Location start, Location end, ArrayList< Location > locations )

Initialized constructor for creating valid search (`Coord` not included)

```

58      {
59          this("8.00", userName, start, end, locations, new ArrayList<Coord>());
60      }

```

#### 6.12.1.4 coolgle.Search.Search ( String startTime, String userName, Location start, Location end, ArrayList< Location > locations, ArrayList< Coord > shortestPath )

Initialized constructor for creating valid search ([Coord](#) included)

```

65         {
66
67             this.startTime = Double.parseDouble(startTime);
68             this.userName = userName;
69             this.start = start;
70             this.end = end;
71             this.locations = locations;
72             this.shortestPath = shortestPath;
73             optimized = false;
74
75         }
```

### 6.12.2 Member Function Documentation

#### 6.12.2.1 void coolgle.Search.addLocation ( Location newLocation )

addLocation Description - Adds the given location to the arraylist of midlocation

```

170         {
171             // Add unless you're reached the limit.
172             if (locations.size() < TRIP_LIMIT)
173                 locations.add(newLocation);
174             else
175                 JOptionPane.showMessageDialog(null, "You have reached the maximun amount of \n"
176                                                 + "locations allowed per trip (" + TRIP_LIMIT + ") "
177                                                 + "Failure", JOptionPane.ERROR_MESSAGE);
178         }
```

#### 6.12.2.2 void coolgle.Search.clearData ( )

clearData Description - Resets all fields, to be used when user clicks the clear search button in the main gui

```

252         {
253             locations.clear();
254             start = new Location();
255             end = new Location();
256             distance = 0;
257             optimized = false;
258             startTime = 8;
259         }
```

#### 6.12.2.3 String coolgle.Search.fileToString ( )

Description - Returns a string that can be printed into the users file.

Additional: The returned string will look something along the lines of " 8 The Kreeger Museum|2401 Foxhall Road Northwest|Washington|District of Columbia|38.9217585|-77.0892007 Fairfax Museum & Visitors Center|10209 Main Street|Fairfax|Virginia|38.8444887|-77.3005546 Thurgood Marshall Airport|7062 Elm Road|Baltimore|-Maryland|39.180862|-76.6675476 " The first location is the starting location, last is ending, middles are mid, ect.. The first number is the start time of the trip Trips will be sorted and valid before they need to be saved.

```

284         {
285             String msg;
286
287             msg = "StartTime " + this.startTime + '\n';
288
289             msg += start.fileToString() + '\n';
290             // Print add all locations
291             Iterator<Location> itrA = locations.iterator();
292             while(itrA.hasNext())
293             {
```

```

294         msg += itrA.next().fileToString() + '\n';
295     }
296     msg += end.fileToString() + '\n';
297
298     msg += "***endSearch" + '\n';
299
300     return msg;
301 }

```

#### 6.12.2.4 Location coolgle.Search.getEnd ( )

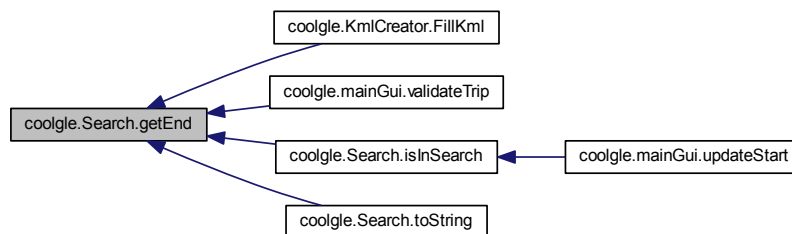
getEnd Description- Returns end location

```

145     {
146         return end;
147     }

```

Here is the caller graph for this function:



#### 6.12.2.5 ArrayList<Location> coolgle.Search.getMidLocations ( )

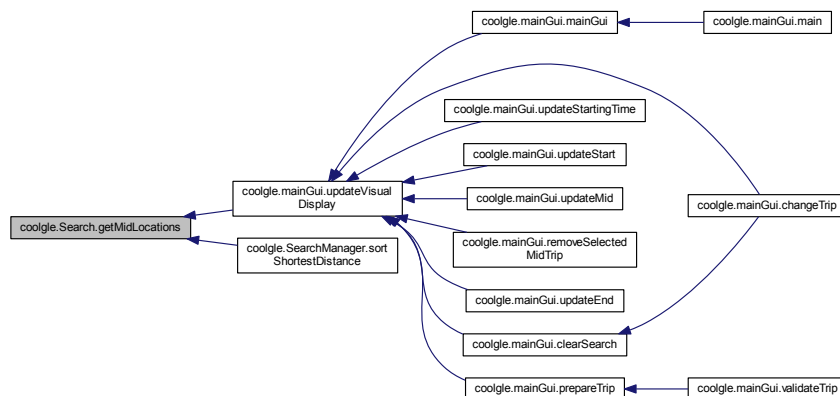
getMidLocations Description - returns arraylist of midlocations.

```

100     {
101         return locations;
102     }

```

Here is the caller graph for this function:



## 6.12.2.6 int coolgle.Search.getNumMidLocations ( )

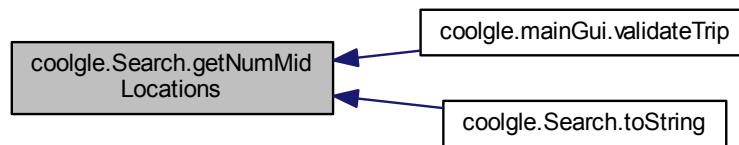
getNumMidLocations Description - Returns the number of mid locations.

```

214         {
215             return locations.size();
216         }

```

Here is the caller graph for this function:



## 6.12.2.7 Location coolgle.Search.getStart ( )

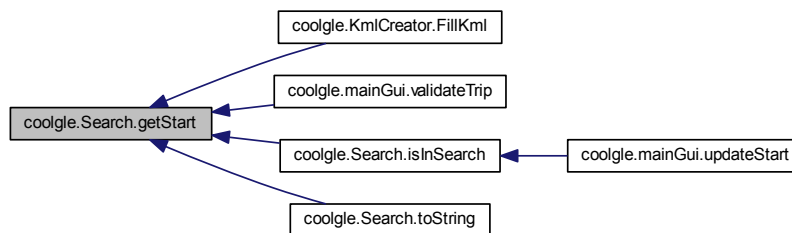
getStart Description - Returns start location.

```

127         {
128             return start;
129         }

```

Here is the caller graph for this function:



## 6.12.2.8 double coolgle.Search.getStartTime ( )

getStartTime Description - Returns start time.

```

118         {
119             return startTime;
120         }

```

Here is the caller graph for this function:



#### 6.12.2.9 String coolgle.Search.getUserName ( )

Description - getUserName

```

161     {
162         return userName;
163     }
  
```

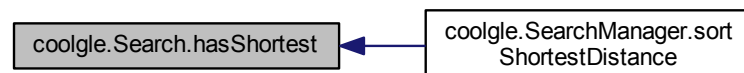
#### 6.12.2.10 boolean coolgle.Search.hasShortest ( )

hasShortest Description- Returns whether or not the search has been optimized.

```

82     {
83         return optimized;
84     }
  
```

Here is the caller graph for this function:



#### 6.12.2.11 boolean coolgle.Search.isInSearch ( Location testLocation )

isInSearch Description - Returns whether or not the given location is in the arraylist of mid locations.

```

224     {
225         boolean alreadyListed = false;
226
227         // Firstly check the start and ending locations
228         if (getStart().isSame(testLocation))
229             alreadyListed = true;
230         else if (getEnd().isSame(testLocation))
231             alreadyListed = true;
232         else
233         {
234             // then the middle locations
235             for (int i = 0; i < locations.size(); i++)
236             {
237                 if (locations.get(i).isSame(testLocation))
238                     alreadyListed = true;
239             }
240         }
241     }
  
```

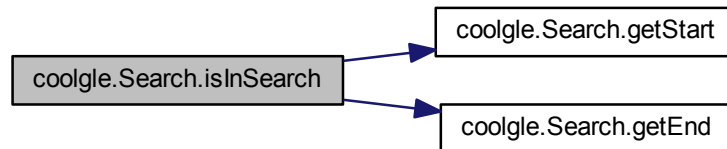


```

240         }
241
242         return alreadyListed;
243     }

```

Here is the call graph for this function:



Here is the caller graph for this function:



#### 6.12.2.12 void coolgle.Search.removeAtPosition ( int *pos* )

`removeAtPosition` Description - Removes the mid location at the given position.

```

205     {
206         locations.remove(pos);
207     }

```

#### 6.12.2.13 boolean coolgle.Search.removeLocation ( Location *removeLocation* )

`removeLocation` Description - Removes the given location from the arraylist of mid locations.

```

185     {
186         //Search through the arraylist, if object exist remove it.
187         boolean removed = false;
188         for ( int i = 0; i < locations.size() && !removed; i++ )
189         {
190             if (locations.get(i).isSame(removeLocation))
191             {
192                 locations.remove(i);
193                 removed = true;
194             }
195         }
196         return removed;
197     }

```

**6.12.2.14 void coolgle.Search.setDistance ( double *distance* )**

setter for distance

```

322     {
323         this.distance = distance;
324     }

```

**6.12.2.15 void coolgle.Search.setEnd ( Location *newEnd* )**

setEnd Description - Sets the end location

```

154     {
155         end = newEnd;
156     }

```

**6.12.2.16 void coolgle.Search.setLocations ( ArrayList< Location > *locations* )**

setter for locations

**Parameters**

<i>locations</i>	
------------------	--

```

307     {
308         this.locations = locations;
309     }

```

**6.12.2.17 void coolgle.Search.setOptimized ( boolean *isOptimized* )**

setOptimized Description - sets optimized to boolean parameter

```

91     {
92         optimized = isOptimized;
93     }

```

**6.12.2.18 void coolgle.Search.setShortestPath ( ArrayList< Coord > *shortestPath* )**

setter for ShortestPath

**Parameters**

<i>locations</i>	
------------------	--

```

315     {
316         this.shortestPath = shortestPath;
317     }

```

**6.12.2.19 void coolgle.Search.setStart ( Location *newStart* )**

setStart Description - Sets the start location

```

136     {
137         start = newStart;
138     }

```

**6.12.2.20 void coolgle.Search.setStartTime ( double *newTime* )**

setStartTime Description - Set start time to new double passed in

```

109         {
110             startTime = newTime;
111         }

```

**6.12.2.21 void coolgle.Search.setUserName ( String *userName* )**

setter for userName

```

329         {
330             this.userName = userName;
331         }

```

**6.12.2.22 String coolgle.Search.toString ( )**

Description - Override of toString

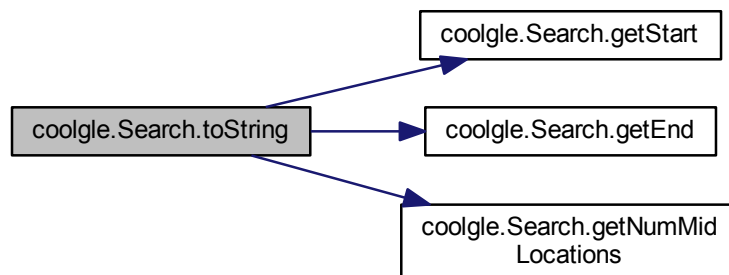
This should be used in the the pervious search dropdown.

```

264         {
265             String returnString = "";
266             returnString += "" + getStart().getName().substring(0, Math.min(
getStart().getName().length(), 16)) + " -> ";
267             returnString += "" + getEnd().getName().substring(0, Math.min(
getEnd().getName().length(), 16)) + " (";
268             returnString += " + " + getNumMidLocations() + " Mid Points) ";
269             return returnString;
270         }

```

Here is the call graph for this function:



The documentation for this class was generated from the following file:

- [src/coolgle/Search.java](#)

**6.13 coolgle.SearchManager Class Reference****Public Member Functions**

- void [sortShortestDistance](#) ([Search](#) search)

## 6.13.1 Member Function Documentation

### 6.13.1.1 void coolgle.SearchManager.sortShortestDistance ( Search search )

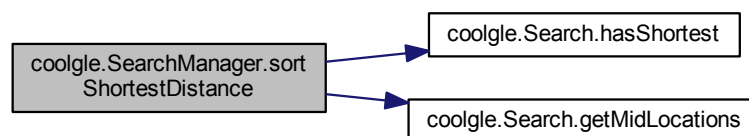
sortShortestDistance Description- need valid search, sort locations start-end with shortest distance

```

25     {
26         //if search already calculated the shortest distance, don't have to go farther
27         if(search.hasShortest()) return;
28         //get start and end locations from search
29         //sort locations start-end with shortest distance
30         ArrayList<Location> temp = new ArrayList<Location>();
31         Location curr = search.getStart();
32         double totalDist = 0.0;
33
34         int size = search.getMidLocations().size();
35
36         for(int i = 0; i < size; i++){
37             double shortestD = Double.MAX_VALUE;
38             int shortest = -1;
39             for(int j = 0; j < search.getMidLocations().size(); j++)
40             {
41                 double dist = curr.distanceTo(search.getMidLocations().get(j));
42                 if((shortest==-1) || dist<shortestD)
43                 {
44                     shortest = j;
45                     shortestD = dist;
46                 }
47             }
48             temp.add(search.getMidLocations().remove(shortest)); //add to end of temp list,
49             removing nearest neighbor from list of candidates
50             totalDist+=shortestD;
51             curr = temp.get(temp.size() - 1);
52         }
53
54         search.setLocations(temp) ;
55         search.setOptimized(true);
56
57         search.setDistance(totalDist + search.getEnd().distanceTo(search.getMidLocations().get(
58             search.getMidLocations().size()-1))); //add distance from last element to ending location.
59     }

```

Here is the call graph for this function:

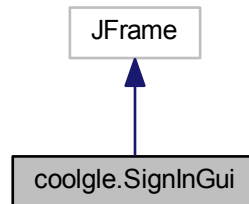


The documentation for this class was generated from the following file:

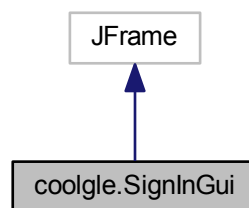
- [src/coolgle/SearchManager.java](#)

## 6.14 coolgle.SignInGui Class Reference

Inheritance diagram for coolgle.SignInGui:



Collaboration diagram for coolgle.SignInGui:



### Public Member Functions

- [SignInGui\(\)](#)

### 6.14.1 Constructor & Destructor Documentation

#### 6.14.1.1 coolgle.SignInGui.SignInGui ( )

Constructor Creates new form SignIn

```

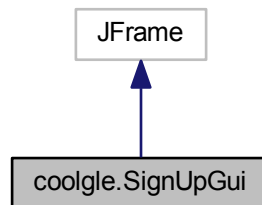
37     {
38         Dimension screen = Toolkit.getDefaultToolkit().getScreenSize();
39         Dimension frm = super.getSize();
40         int xpos = (int) (screen.getWidth() / 8 - frm.getWidth() / 2);
41         int ypos = (int) (screen.getHeight() / 8 - frm.getHeight() / 2);
42         super.setLocation(xpos, ypos);
43
44         initComponents();
45
46         pswdText.setText("");
47     }
  
```

The documentation for this class was generated from the following file:

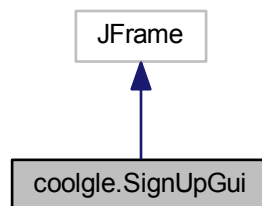
- [src/coolgle/SignInGui.java](#)

## 6.15 coolgle.SignUpGui Class Reference

Inheritance diagram for coolgle.SignUpGui:



Collaboration diagram for coolgle.SignUpGui:



### Public Member Functions

- [SignUpGui\(\)](#)

### 6.15.1 Constructor & Destructor Documentation

#### 6.15.1.1 coolgle.SignUpGui.SignUpGui ( )

Constructor Creates new form SignUp

```
42     {  
43         Dimension screen = Toolkit.getDefaultToolkit().getScreenSize();  
44         Dimension frm = super.getSize();  
45         int xpos = (int) (screen.getWidth() / 8 - frm.getWidth() / 2);  
46         int ypos = (int) (screen.getHeight() / 8 - frm.getHeight() / 2);  
47         super.setLocation(xpos, ypos);  
48  
49         initComponents();  
50     }
```

The documentation for this class was generated from the following file:

- [src/coolgle/SignUpGui.java](#)

## 6.16 coolgle.UIManager Class Reference

### Public Member Functions

- [UIManager](#) ()

### 6.16.1 Constructor & Destructor Documentation

#### 6.16.1.1 coolgle.UIManager.UIManager ( )

```
20      {
21          displayLogInGui();
22      }
```

The documentation for this class was generated from the following file:

- [src/coolgle/UIManager.java](#)

## 6.17 coolgle.UserAuthentication Class Reference

### Public Member Functions

- boolean [logUser](#) (String userName, char[] password, int type)
- boolean [addUser](#) (String name, String pass, String email)
- boolean [addUserWithType](#) (String name, String pass, String email, int userType)
- void [removeUser](#) (String userName, boolean removeFile)
- String [getEmail](#) (String userName)
- String [getPassword](#) (String userName)
- int [getType](#) (String userName)
- ArrayList< String > [getNames](#) ()
- void [updateUser](#) (String oldUserName, String userName, String userPass, String userEmail, int userType)
- boolean [userExists](#) (String userName)

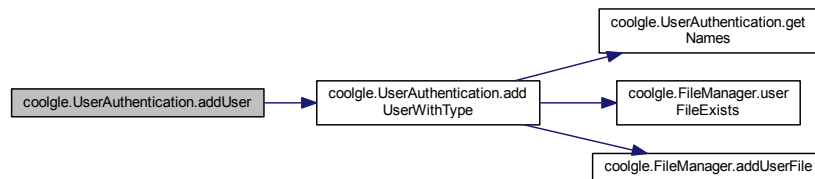
### 6.17.1 Member Function Documentation

#### 6.17.1.1 boolean coolgle.UserAuthentication.addUser ( String name, String pass, String email )

**addUser** Description - Given a new un / pw / email adds the user to the list of users Note : All users are created as users, an admin must promote them to admin returns true if user was successfully created. Note : Also needs to create a user File to store recent searches, named same as username Pre- Post -

```
83      {
84          return addUserWithType(name, pass, email, 0);
85      }
```

Here is the call graph for this function:



#### 6.17.1.2 boolean coolgle.UserAuthentication.addUserWithType ( String name, String pass, String email, int userType )

addUserWithType

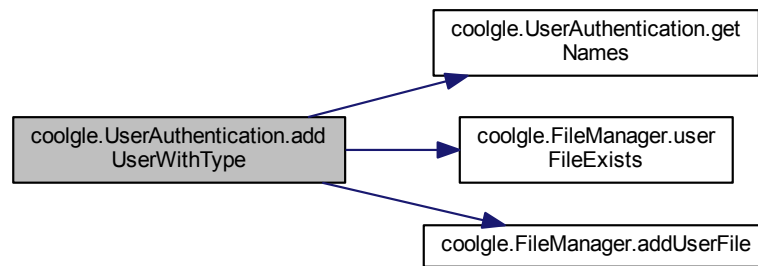
Description - Given a new un / pw / email adds the user to the list of users returns true if user was successfully created. Note : Also needs to create a user File to store recent searches, named same as username Shows an error if you are only the user limit. Pre- Post -

```

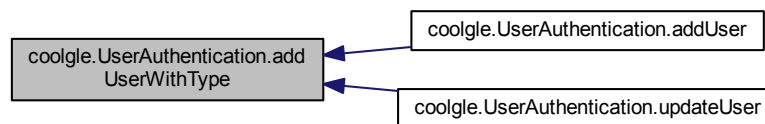
96     {
97         boolean addedSuccessfully = true;
98
99         if (getNames().size() <= MAX_USERS)
100         {
101             BufferedWriter writer = null;
102             try
103             {
104                 String newUser = name + "|" + pass + "|" + email + "|" + userType;
105                 writer = new BufferedWriter(new FileWriter(USERFILE, true));
106                 writer.append(newUser);
107                 writer.newLine();
108                 writer.close();
109             }
110             catch (IOException ex)
111             {
112                 addedSuccessfully = false;
113             }
114
115             // Then create the user's data file, but only they don't already have one:
116             FileManager fm = new FileManager();
117             if (!fm.userFileExists(name)) fm.addUserFile(name);
118         }
119         else
120         {
121             JOptionPane.showMessageDialog(null, "Maximun number of users reached", "Failure",
122             JOptionPane.ERROR_MESSAGE);
123         }
124         return addedSuccessfully;
125     }
  
```



Here is the call graph for this function:



Here is the caller graph for this function:



### 6.17.1.3 String coolgle.UserAuthentication.getEmail ( String userName )

getEmail

Description- Given user name, return the users email Pre- Post -

```

202     {
203         //Spin through file, find user, grab email section :
204         String returnEmail = "";
205         try
206         {
207             BufferedReader br = new BufferedReader(new FileReader(USERFILE));
208             String userString;
209             while ((userString = br.readLine()) != null)
210             {
211                 String[] data = userString.split("\\|");
212                 if ( data[0].compareTo(userName) == 0 )
213                     returnEmail = data[2];
214             }
215             br.close();
216         }
217         catch (IOException ex)
218         {
219             Logger.getLogger(UserAuthentication.class.getName()).log(Level.SEVERE, null, ex);
220         }
221         return returnEmail;
222     }
223 
```

### 6.17.1.4 ArrayList<String> coolgle.UserAuthentication.getNames ( )

ArrayList

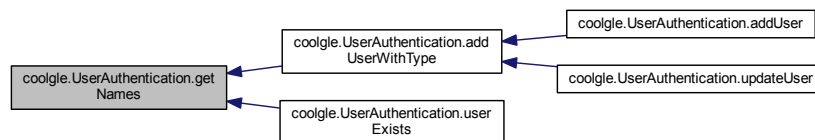
Description - Returns an array list of all the user names. Pre- Post -

```

292     {
293         ArrayList<String> names = new ArrayList<String>();
294         try
295         {
296             BufferedReader br = new BufferedReader(new FileReader(USERFILE));
297             String userString;
298             while ((userString = br.readLine()) != null)
299             {
300                 String[] data = userString.split("\\|");
301                 String un = data[0];
302                 names.add(un);
303             }
304             br.close();
305         }
306         catch (IOException ex)
307         {
308             Logger.getLogger(UserAuthentication.class.getName()).log(Level.SEVERE, null, ex);
309         }
310
311         Collections.sort(names); //Get names in sorted order
312
313         return names;
314     }

```

Here is the caller graph for this function:



#### 6.17.1.5 String coolgle.UserAuthentication.getPassword ( String userName )

getPassword

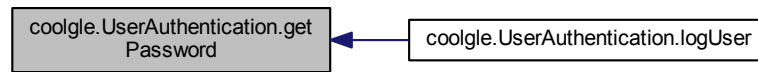
Description- Given user name, return the users password Pre- Post -

```

231     {
232         //Spin through file, find user, grab password section :
233         String returnPass = "";
234         try
235         {
236             BufferedReader br = new BufferedReader(new FileReader(USERFILE));
237             String userString;
238             while ((userString = br.readLine()) != null)
239             {
240                 String[] data = userString.split("\\|");
241                 if ( data[0].compareTo(userName) == 0 )
242                     returnPass = data[1];
243             }
244             br.close();
245         }
246         catch (IOException ex)
247         {
248             Logger.getLogger(UserAuthentication.class.getName()).log(Level.SEVERE, null, ex);
249         }finally{
250
251         }
252
253         return returnPass;
254     }

```

Here is the caller graph for this function:



#### 6.17.1.6 int coolgle.UserAuthentication.getType ( String userName )

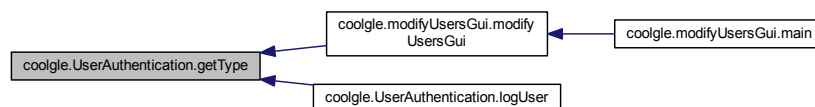
getType

Description- Given user name, returns the users is a user (0) or admin (1) Pre- Post -

```

262     {
263         //Spin through file, find user, grab password section :
264         String returnType = "";
265         try
266         {
267             BufferedReader br = new BufferedReader(new FileReader(USERFILE));
268             String userString;
269             while ((userString = br.readLine()) != null)
270             {
271                 String[] data = userString.split("\\|");
272                 if ( data[0].compareTo(userName) == 0 )
273                     returnType = data[3];
274             }
275             br.close();
276         }
277         catch (IOException ex)
278         {
279             Logger.getLogger(UserAuthentication.class.getName()).log(Level.SEVERE, null, ex);
280         }
281         return Integer.parseInt(returnType);
282     }
283 
```

Here is the caller graph for this function:



#### 6.17.1.7 boolean coolgle.UserAuthentication.logUser ( String userName, char[] password, int type )

constructor for [UserAuthentication](#) object

```
public UserAuthentication() {
```

} logUser Description - Given a UN / Password and userType, checks to see the credentials are correct. Note : passwords are stored as char[] Note, user types are 1 = admin, 0 = user

```

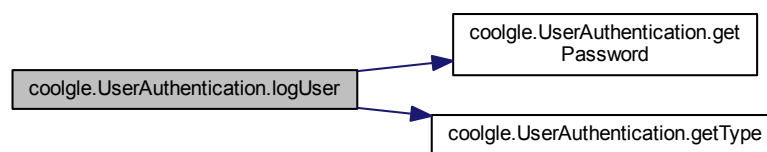
55     {
56         //Get string for password :
57         String pass = new String(password);
  
```

```

58         boolean login = false;
59
60         if (pass.compareTo(getPassword(userName)) == 0)
61         {
62             if (!(getType(userName) == 0 && type == 1)) //Checks to see if a user tries to log
in as an admin
63                 login = true;
64         }
65
66         if ( userName.compareTo("admin") == 0 && pass.compareTo("admin") == 0)
67         {
68             //Always allow the combination admin, admin to work
69             login = true;
70         }
71         return login;
72     }

```

Here is the call graph for this function:



#### 6.17.1.8 void coolgle.UserAuthentication.removeUser ( String userName, boolean removeFile )

**removeUser Description-** Given the userName, removes them from the file of users Also, removes their file, if removeFile parameter is true. Pre- Post -

```

136     {
137         // Create a arraylist of all the users except for the one being removed.
138         ArrayList<String> users = new ArrayList<String>();
139         try
140         {
141             BufferedReader br = new BufferedReader(new FileReader(USERFILE));
142
143             String userString;
144             while ((userString = br.readLine()) != null)
145             {
146                 String[] data = userString.split("\\|");
147                 if ( data[0].compareTo(userName) != 0 ) // This Line needs to be written to new file.
148                 {
149                     users.add(userString);
150                 }
151                 else
152                 {
153                     //This is the user we want to detele, don't reprint.
154                 }
155             }
156             br.close();
157         }
158         catch (IOException ex)
159         {
160             Logger.getLogger(UserAuthentication.class.getName()).log(Level.SEVERE, null, ex);
161         }
162
163         // Clear file and reAdd users
164         BufferedWriter clearFile = null;
165         BufferedWriter rewriteFile = null;
166         try
167         {
168             clearFile = new BufferedWriter(new FileWriter(USERFILE, false));
169             clearFile.append("");
170             clearFile.close();
171
172             rewriteFile = new BufferedWriter(new FileWriter(USERFILE, true));
173
174             for (int i = 0; i < users.size(); i++)

```

```

175         {
176             rewriteFile.append(users.get(i));
177             rewriteFile.newLine();
178         }
179
180         rewriteFile.close();
181     }
182     catch (IOException ex)
183     {
184         //Err
185     }
186
187     // Delete File
188     if (removeFile)
189     {
190         FileManager fileDel = new FileManager();
191         fileDel.removeUserFile(userName);
192     }
193 }

```

Here is the caller graph for this function:



#### 6.17.1.9 void coolgle.UserAuthentication.updateUser ( String *oldUserName*, String *userName*, String *userPass*, String *userEmail*, int *userType* )

updateUser

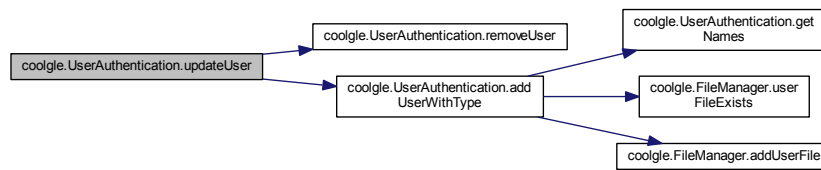
Description - Returns an arraylist of all the user names. Pre- Post -

```

322     {
323         try
324         {
325             BufferedReader br = new BufferedReader(new FileReader(USERFILE));
326             String userString;
327             while ((userString = br.readLine()) != null)
328             {
329                 String[] data = userString.split("\\|");
330                 String un = data[0];
331                 if ( un.compareTo(oldUserName) == 0 ) // This is the user we want to edit
332                 {
333                     // Remove the user from the file
334                     removeUser(oldUserName, false); // Just remove user, NOT their file
335                     // Read them to the end
336                     addUserWithType(userName, userPass, userEmail, userType);
337                 }
338             }
339             br.close();
340         }
341         catch (IOException ex)
342         {
343             Logger.getLogger(UserAuthentication.class.getName()).log(Level.SEVERE, null, ex);
344         }
345     }

```

Here is the call graph for this function:



#### 6.17.1.10 boolean coolgle.UserAuthentication.userExists ( String *userName* )

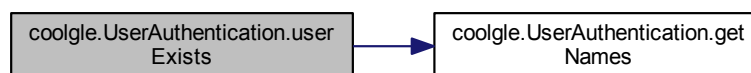
`userExists`

Description - Returns true if username is already taken Post - Returns if the username is already taken

```

353     {
354         ArrayList<String> names = getNames();
355         boolean exists = false;
356
357         if (names.contains(userName) || userName.compareTo("admin") == 0 ) // Note : admin is a resv.
358             name         exists = true;
359
360         return exists;
361     }
  
```

Here is the call graph for this function:

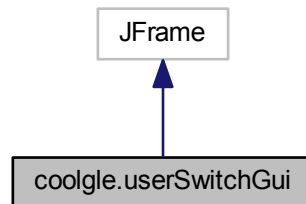


The documentation for this class was generated from the following file:

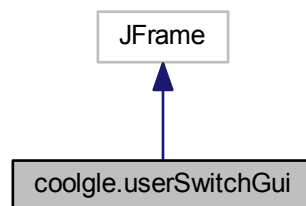
- `src/coolgle/UserAuthentication.java`

## 6.18 coolgle.userSwitchGui Class Reference

Inheritance diagram for coolgle.userSwitchGui:



Collaboration diagram for coolgle.userSwitchGui:



### Public Member Functions

- `userSwitchGui (adminGui m_main, String userName)`

#### 6.18.1 Constructor & Destructor Documentation

##### 6.18.1.1 coolgle.userSwitchGui.userSwitchGui ( adminGui m\_main, String userName )

Constructor Creates new form userSwitch

```

40     {
41         Dimension screen = Toolkit.getDefaultToolkit().getScreenSize();
42         Dimension frm = super.getSize();
43         int xpos = (int) (screen.getWidth() / 8 - frm.getWidth() / 2);
44         int ypos = (int) (screen.getHeight() / 8 - frm.getHeight() / 2);
45         super.setLocation(xpos, ypos);
46
47         // Makes it so that you can exit from pop up messages and this window without closing the entire
48         app
49         this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
50         user = userName;
51         mainFrame = m_main;
52     }
  
```

```
53         initComponents();  
54     }
```

The documentation for this class was generated from the following file:

- [src/coolgle/userSwitchGui.java](#)



## Chapter 7

# File Documentation

### 7.1 src/coolgle/AddLocationGui.java File Reference

#### Classes

- class [coolgle.AddLocationGui](#)

#### Packages

- package [coolgle](#)

### 7.2 src/coolgle/adminGui.java File Reference

#### Classes

- class [coolgle.adminGui](#)

#### Packages

- package [coolgle](#)

### 7.3 src/coolgle/adminSwitchGui.java File Reference

#### Classes

- class [coolgle.adminSwitchGui](#)

#### Packages

- package [coolgle](#)

### 7.4 src/coolgle/Coord.java File Reference

### Classes

- class [coolgle.Coord](#)

### Packages

- package [coolgle](#)

## 7.5 src/coolgle/Driver.java File Reference

### Classes

- class [coolgle.Driver](#)

### Packages

- package [coolgle](#)

## 7.6 src/coolgle/FileManager.java File Reference

### Classes

- class [coolgle.FileManager](#)

### Packages

- package [coolgle](#)

## 7.7 src/coolgle/Geocoder.java File Reference

### Classes

- class [coolgle.Geocoder](#)

### Packages

- package [coolgle](#)

## 7.8 src/coolgle/KmlCreator.java File Reference

### Classes

- class [coolgle.KmlCreator](#)

### Packages

- package [coolgle](#)

## 7.9 src/coolgle/Location.java File Reference

### Classes

- class [coolgle.Location](#)

### Packages

- package [coolgle](#)

## 7.10 src/coolgle/mainGui.java File Reference

### Classes

- class [coolgle.mainGui](#)

### Packages

- package [coolgle](#)

## 7.11 src/coolgle/modifyUsersGui.java File Reference

### Classes

- class [coolgle.modifyUsersGui](#)

### Packages

- package [coolgle](#)

## 7.12 src/coolgle/Search.java File Reference

### Classes

- class [coolgle.Search](#)

### Packages

- package [coolgle](#)

## 7.13 src/coolgle/SearchManager.java File Reference

### Classes

- class [coolgle.SearchManager](#)

## Packages

- package [coolgle](#)

## 7.14 src/coolgle/SignInGui.java File Reference

### Classes

- class [coolgle.SignInGui](#)

## Packages

- package [coolgle](#)

## 7.15 src/coolgle/SignUpGui.java File Reference

### Classes

- class [coolgle.SignUpGui](#)

## Packages

- package [coolgle](#)

## 7.16 src/coolgle/UiManager.java File Reference

### Classes

- class [coolgle.UiManager](#)

## Packages

- package [coolgle](#)

## 7.17 src/coolgle/UserAuthentication.java File Reference

### Classes

- class [coolgle.UserAuthentication](#)

## Packages

- package [coolgle](#)

## 7.18 src/coolgle/userSwitchGui.java File Reference

### Classes

- class [coolgle.userSwitchGui](#)

### Packages

- package [coolgle](#)

# Index

- addLocation
  - coolgle::Search, [61](#)
- AddLocationGui
  - coolgle::AddLocationGui, [12](#), [13](#)
- addSearch
  - coolgle::FileManager, [24](#)
- addUser
  - coolgle::UserAuthentication, [71](#)
- addUserFile
  - coolgle::FileManager, [24](#)
- addUserWithType
  - coolgle::UserAuthentication, [72](#)
- adminGui
  - coolgle::adminGui, [15](#)
- adminSwitchGui
  - coolgle::adminSwitchGui, [21](#)
- changeTrip
  - coolgle::mainGui, [46](#)
- clearData
  - coolgle::Search, [61](#)
- clearSearch
  - coolgle::mainGui, [47](#)
- coolgle, [9](#)
- coolgle.AddLocationGui, [11](#)
- coolgle.adminGui, [14](#)
- coolgle.adminSwitchGui, [20](#)
- coolgle.Coord, [21](#)
- coolgle.Driver, [23](#)
- coolgle.FileManager, [23](#)
- coolgle.Geocoder, [28](#)
- coolgle.KmlCreator, [30](#)
- coolgle.Location, [33](#)
- coolgle.mainGui, [44](#)
- coolgle.modifyUsersGui, [57](#)
- coolgle.Search, [59](#)
- coolgle.SearchManager, [67](#)
- coolgle.SignInGui, [69](#)
- coolgle.SignUpGui, [70](#)
- coolgle.UIManager, [71](#)
- coolgle.UserAuthentication, [71](#)
- coolgle.userSwitchGui, [79](#)
- coolgle::AddLocationGui
  - AddLocationGui, [12](#), [13](#)
  - setUpGui, [13](#)
- coolgle::Coord
  - Coord, [21](#), [22](#)
  - getLatitude, [22](#)
  - getLongitude, [22](#)
  - setLatitude, [22](#)
  - setLongitude, [23](#)
  - toString, [23](#)
- coolgle::Driver
  - main, [23](#)
- coolgle::FileManager
  - addSearch, [24](#)
  - addUserFile, [24](#)
  - getPrev, [25](#)
  - isFull, [26](#)
  - removeUserFile, [27](#)
  - userFileExists, [27](#)
- coolgle::Geocoder
  - createLocation, [28](#)
- coolgle::KmlCreator
  - FillKml, [30](#)
  - KmlCreator, [30](#)
  - openKml, [31](#)
  - printTime, [32](#)
- coolgle::Location
  - distanceTo, [35](#)
  - fileToString, [35](#)
  - getAddress, [36](#)
  - getCity, [36](#)
  - getCoord, [37](#)
  - getLatitude, [37](#)
  - getLongitude, [37](#)
  - getName, [38](#)
  - getState, [38](#)
  - isSame, [39](#)
  - Location, [33](#), [34](#)
  - printToFile, [39](#), [40](#)
  - setAddress, [41](#)
  - setCity, [41](#)
  - setCoord, [41](#)
  - setLatitude, [42](#)
  - setLongitude, [42](#)
  - setName, [42](#)
  - setState, [43](#)
  - toString, [43](#)
- coolgle::Search
  - addLocation, [61](#)
  - clearData, [61](#)
  - fileToString, [61](#)
  - getEnd, [62](#)
  - getMidLocations, [62](#)
  - getNumMidLocations, [62](#)
  - getStart, [63](#)
  - getStartTime, [63](#)
  - getUserName, [64](#)

- hasShortest, [64](#)
- isInSearch, [64](#)
- removeAtPosition, [65](#)
- removeLocation, [65](#)
- Search, [60](#)
- setDistance, [65](#)
- setEnd, [66](#)
- setLocations, [66](#)
- setOptimized, [66](#)
- setShortestPath, [66](#)
- setStart, [66](#)
- setStartTime, [66](#)
- setUserName, [67](#)
- toString, [67](#)
- coolgle::SearchManager
  - sortShortestDistance, [68](#)
- coolgle::SignInGui
  - SignInGui, [69](#)
- coolgle::SignUpGui
  - SignUpGui, [70](#)
- coolgle::UIManager
  - UIManager, [71](#)
- coolgle::UserAuthentication
  - addUser, [71](#)
  - addUserWithType, [72](#)
  - getEmail, [73](#)
  - getNames, [73](#)
  - getPassword, [74](#)
  - getType, [75](#)
  - logUser, [75](#)
  - removeUser, [76](#)
  - updateUser, [77](#)
  - userExists, [78](#)
- coolgle::adminGui
  - adminGui, [15](#)
  - delUser, [16](#)
  - editLocation, [17](#)
  - logOut, [17](#)
  - modUser, [17](#)
  - populateLocationList, [18](#)
  - populateUserList, [18](#)
  - recreateDatabase, [19](#)
- coolgle::adminSwitchGui
  - adminSwitchGui, [21](#)
- coolgle::mainGui
  - changeTrip, [46](#)
  - clearSearch, [47](#)
  - logOut, [48](#)
  - main, [48](#)
  - mainGui, [45](#)
  - populateLocationList, [49](#)
  - prepareTrip, [50](#)
  - removeSelectedMidTrip, [51](#)
  - updateEnd, [52](#)
  - updateMid, [52](#)
  - updateStart, [53](#)
  - updateStartingTime, [53](#)
  - updateVisualDisplay, [54](#)
  - validateTrip, [55](#)
  - viewHelp, [56](#)
  - viewInformation, [56](#)
- coolgle::modifyUsersGui
  - main, [58](#)
  - modifyUsersGui, [57](#)
- coolgle::userSwitchGui
  - userSwitchGui, [79](#)
- Coord
  - coolgle::Coord, [21](#), [22](#)
- createLocation
  - coolgle::Geocoder, [28](#)
- delUser
  - coolgle::adminGui, [16](#)
- distanceTo
  - coolgle::Location, [35](#)
- editLocation
  - coolgle::adminGui, [17](#)
- fileToString
  - coolgle::Location, [35](#)
  - coolgle::Search, [61](#)
- FillKml
  - coolgle::KmlCreator, [30](#)
- getAddress
  - coolgle::Location, [36](#)
- getCity
  - coolgle::Location, [36](#)
- getCoord
  - coolgle::Location, [37](#)
- getEmail
  - coolgle::UserAuthentication, [73](#)
- getEnd
  - coolgle::Search, [62](#)
- getLatitude
  - coolgle::Coord, [22](#)
  - coolgle::Location, [37](#)
- getLongitude
  - coolgle::Coord, [22](#)
  - coolgle::Location, [37](#)
- getMidLocations
  - coolgle::Search, [62](#)
- getName
  - coolgle::Location, [38](#)
- getNames
  - coolgle::UserAuthentication, [73](#)
- getNumMidLocations
  - coolgle::Search, [62](#)
- getPassword
  - coolgle::UserAuthentication, [74](#)
- getPrev
  - coolgle::FileManager, [25](#)
- getStart
  - coolgle::Search, [63](#)
- getStartTime
  - coolgle::Search, [63](#)

- getState
  - coolgle::Location, 38
- getType
  - coolgle::UserAuthentication, 75
- getUserName
  - coolgle::Search, 64
- hasShortest
  - coolgle::Search, 64
- isFull
  - coolgle::FileManager, 26
- isInSearch
  - coolgle::Search, 64
- isSame
  - coolgle::Location, 39
- KmlCreator
  - coolgle::KmlCreator, 30
- Location
  - coolgle::Location, 33, 34
- logOut
  - coolgle::adminGui, 17
  - coolgle::mainGui, 48
- logUser
  - coolgle::UserAuthentication, 75
- main
  - coolgle::Driver, 23
  - coolgle::mainGui, 48
  - coolgle::modifyUsersGui, 58
- mainGui
  - coolgle::mainGui, 45
- modUser
  - coolgle::adminGui, 17
- modifyUsersGui
  - coolgle::modifyUsersGui, 57
- openKml
  - coolgle::KmlCreator, 31
- populateLocationList
  - coolgle::adminGui, 18
  - coolgle::mainGui, 49
- populateUserList
  - coolgle::adminGui, 18
- prepareTrip
  - coolgle::mainGui, 50
- printTime
  - coolgle::KmlCreator, 32
- printToFile
  - coolgle::Location, 39, 40
- recreateDatabase
  - coolgle::adminGui, 19
- removeAtPosition
  - coolgle::Search, 65
- removeLocation
  - coolgle::Search, 65
- removeSelectedMidTrip
  - coolgle::mainGui, 51
- removeUser
  - coolgle::UserAuthentication, 76
- removeUserFile
  - coolgle::FileManager, 27
- Search
  - coolgle::Search, 60
- setAddress
  - coolgle::Location, 41
- setCity
  - coolgle::Location, 41
- setCoord
  - coolgle::Location, 41
- setDistance
  - coolgle::Search, 65
- setEnd
  - coolgle::Search, 66
- setLatitude
  - coolgle::Coord, 22
  - coolgle::Location, 42
- setLocations
  - coolgle::Search, 66
- setLongitude
  - coolgle::Coord, 23
  - coolgle::Location, 42
- setName
  - coolgle::Location, 42
- setOptimized
  - coolgle::Search, 66
- setShortestPath
  - coolgle::Search, 66
- setStart
  - coolgle::Search, 66
- setStartTime
  - coolgle::Search, 66
- setState
  - coolgle::Location, 43
- setUpGui
  - coolgle::AddLocationGui, 13
- setUserName
  - coolgle::Search, 67
- SignInGui
  - coolgle::SignInGui, 69
- SignUpGui
  - coolgle::SignUpGui, 70
- sortShortestDistance
  - coolgle::SearchManager, 68
- src/coolgle/AddLocationGui.java, 81
- src/coolgle/Coord.java, 81
- src/coolgle/Driver.java, 82
- src/coolgle/FileManager.java, 82
- src/coolgle/Geocoder.java, 82
- src/coolgle/KmlCreator.java, 82
- src/coolgle/Location.java, 83
- src/coolgle/Search.java, 83
- src/coolgle/SearchManager.java, 83
- src/coolgle/SignInGui.java, 84



- src/coolgle/SignUpGui.java, [84](#)
- src/coolgle/UiManager.java, [84](#)
- src/coolgle/UserAuthentication.java, [84](#)
- src/coolgle/adminGui.java, [81](#)
- src/coolgle/adminSwitchGui.java, [81](#)
- src/coolgle/mainGui.java, [83](#)
- src/coolgle/modifyUsersGui.java, [83](#)
- src/coolgle/userSwitchGui.java, [85](#)

#### toString

- coolgle::Coord, [23](#)
- coolgle::Location, [43](#)
- coolgle::Search, [67](#)

#### UiManager

- coolgle::UiManager, [71](#)

#### updateEnd

- coolgle::mainGui, [52](#)

#### updateMid

- coolgle::mainGui, [52](#)

#### updateStart

- coolgle::mainGui, [53](#)

#### updateStartingTime

- coolgle::mainGui, [53](#)

#### updateUser

- coolgle::UserAuthentication, [77](#)

#### updateVisualDisplay

- coolgle::mainGui, [54](#)

#### userExists

- coolgle::UserAuthentication, [78](#)

#### userFileExists

- coolgle::FileManager, [27](#)

#### userSwitchGui

- coolgle::userSwitchGui, [79](#)

#### validateTrip

- coolgle::mainGui, [55](#)

#### viewHelp

- coolgle::mainGui, [56](#)

#### viewInformation

- coolgle::mainGui, [56](#)