

Class-Responsibility-Collaboration (CRC) Cards

Class: MainWindowClass
Responsibility: <ul style="list-style-type: none">1. Send message to ContactManagerClass to open.2. Send message to ReportManagerClass to open.3. Send message to SettingsManagerClass to open.4. Generate a (unique)instanceUUID value.5. Encapsulates all other classes.6. Encapsulates all application level information.
Collaboration: <ul style="list-style-type: none">1. ContactManagerClass(Dependency)2. ReportManagerClass(Dependency)3. SettingsManagerClass (Dependency)

Class: ReportManagerClass

Responsibility:

1. Check with **MainWindowClass** if database is ready.
2. Send request for reports data to **DatabaseConnectorClass**.
3. Receive reports data from **DatabaseConnectorClass**.
4. Make a new **ReportClass** for each possible report.
5. Send message to **AnalyticsConnectorClass** class requesting for data.
6. Send message to **ContactManagerClass** to request data.
7. Send message to **MailManagerClass** to start.
8. Process received data into a series of ready reports.
9. Binds received contacts to ready-made report(s).
10. Sends a request for contents of a department to **ContactManagerClass**.
11. Responsible for all report related functionality.
12. Save new **ReportClass** instances into database.
13. Respond to changes in related **ReportClass** Instances.
14. Process all necessary **ReportClass** report results into **EmailMessageClass** instances.
15. Push **EmailMessageClass** instances to **MailManagerClass**.

Collaboration:

1. **AnalyticsConnectorClass**(Dependency)
2. **DatabaseConnectorClass**(Dependency)
3. **ContactManagerClass**
4. **EmailMessageClass**
5. **MailManagerClass**
6. **MainWindowClass**
7. **ReportClass**

Class: **ContactManagerClass**

Responsibility:

1. Send message to **DatabaseConnectorClass** to open (if not opened).
2. Send queries to **DatabaseConnectorClass**.
3. Receive data from **DatabaseConnectorClass**.
4. Make a new **ContactClass** for each entry received from **DatabaseConnectorClass**.
5. Serialize all **ContactClass** instances and send to **DatabaseConnectorClass**.
6. Make a new **DepartmentClass** for each entry received from **DatabaseConnectorClass**.
7. Serialize all **DepartmentClass** instances and send to **DatabaseConnectorClass**.

Collaboration:

1. **DatabaseConnectorClass**(Dependency)
2. **ContactClass**
3. **DepartmentClass**

Class: **SettingsManagerClass**

Responsibility:

1. Send requests for settings data **FileConnectorClass**.
2. Process incoming data from **FileConnectorClass**.
3. Process requests from **MailManagerClass** for settings.
4. Process requests from **DatabaseConnectorClass** for settings.
5. Manage **MailManagerSettingsClass** and **DatabaseSettingsClass** using settings data.
6. Send requests to **FileConnectorClass** to save settings data.

Collaboration:

1. **FileConnectorClass** (Dependency)
2. **DatabaseSettingsClass**
3. **MailManagerSettingsClass**
4. **MainWindowClass**

Class: **ReportClass**

Responsibility:

1. Respond to changes in the report data from **ReportManagerClass**.
2. Notify **ReportManagerClass** when it is safe to write changes back to database.
3. Return necessary parameters for a given report as request by **ReportManagerClass**.

Collaboration:

1. **ReportManagerClass**

Class: **DepartmentClass**

Responsibility:

1. Hold data for department-related information.
2. Respond to requests to various pieces of itself from **ContactManagerClass**.
3. Respond to changes to itself as requested by **ContactManagerClass**.
4. Notify **ContactManagerClass** when changes are successful and can be written to database.
5. Department should directory store all related contact information.

Collaboration:

1. **ContactManagerClass**

Class: **ContactClass**

Responsibility:

1. Respond to requests for information from **ContactManagerClass**.
2. Process inbound changes as requested by **ContactManagerClass**.
3. Notify **ContactManagerClass** when it is safe to write changes to database.
4. Notify **ContactManagerClass** when data for the requested information does not exist.
5. References **DepartmentAssociationClass** to determine contact-department associations.

Collaboration:

1. **ContactManagerClass**
2. **DepartmentAssociationClass**

Class: **EmailMessageClass**

Responsibility:

1. Encapsulate all data related to each email message.
2. Receive change requests from **ReportManagerClass**.
3. Send message to **MailManagerClass** when it is ready to be sent.

Collaboration:

1. **MailManagerClass**
2. **ReportManagerClass**

Class: MailManagerSettingsClass
Responsibility: <ol style="list-style-type: none"> 1. Respond to request for information from SettingsManagerClass. 2. Process inbound changes as requested by SettingsManagerClass.
Collaboration: <ol style="list-style-type: none"> 1. SettingsManagerClass

Class: DatabaseSettingsClass
Responsibility: <ol style="list-style-type: none"> 1. Respond to request for information from SettingsManagerClass. 2. Process inbound changes as requested by SettingsManagerClass.
Collaboration: <ol style="list-style-type: none"> 1. SettingsManagerClass

Class: MailManagerClass
Responsibility: <ol style="list-style-type: none"> 1. Recieve EmailMessageClass from ReportManagerClass. 2. Request settings from SettingsManagerClass. 3. Send emails through Mail Server by reading each EmailMessageClass instance.
Collaboration: <ol style="list-style-type: none"> 1. SettingsManagerClass (Dependency) 2. EmailMessageClass 3. ReportManagerClass 4. Mail Server (External)

Class: AnalyticsConnectorClass

Responsibility:

1. Send requests for data to **Google Analytics**.
2. Process incoming data from **Google Analytics**.
3. Receive requests from **ReportManagerClass** and translate into requests ready to be sent to **Google Analytics**.
4. Send received to **ReportManagerClass**.
5. Notify **ReportManagerClass** when data is ready to be retrieved.

Collaboration:

1. **ReportManagerClass**
2. **Google Analytics** (External)

Class: **DatabaseConnectorClass**

Responsibility:

1. Request settings from **SettingsManagerClass**.
2. Connect to database management system.
3. Handles all database-related traffic.
4. Translate queries from **ReportManagerClass** into database queries.
5. Translate queries from **ContactManagerClass** into database queries.

Collaboration:

1. **SettingsManagerClass**(Dependency)
2. **ContactManagerClass**
3. **ReportManagerClass**

Class: **FileConnectorClass**

Responsibility:

1. Load or save settings file in filesystem.
2. Receive requests from **SettingsManagerClass** to load or save settings file.
3. Notify **SettingsManagerClass** when settings file cannot be found.
4. Notify **SettingsManagerClass** when settings file is ready to be retrieved.

Collaboration:

1. **SettingsManagerClass**