

Glossary

General		
	Add	Displays a Pop-up Window of available Contacts to add to the current Department or a list of Contacts and Departments to add to a Report. Only visible when "Edit" has been selected for the current Department or Report.
	Administrator	Main user of the GUI.
	Cancel	Reverts the changes made to Editable Fields.
	Confirmation Message	The Pop-up Window displayed when confirming saves, deletions, and sending of Reports.
	Contact	Information within the system pertaining to an individual, such as address, phone number, email address, and other Fields.
	Core Reporting API	A means of accessing and manipulating the data from GA.
	Create	Shown as a "+" sign on the GUI. Creates a new item to the current List Pane.
	Database Error	Any error that may be returned by the database.
	Delete	Shown as a trash bin on the GUI. Deletes the selected item in the current List Pane.
	Department	A collection of Employee Contacts grouped into departments within the company.
	Department Association	A database table in which each row is a link between a Department and a Contact.
	Details Pane	A section of the GUI for viewing or manipulating a single Contact, Department, or Report Profile.
	Dimensions*	Breaks down metrics across some common criteria, such as date, city, browser, and operating system.
	Edit	Makes the Fields on the current Details Pane Editable Fields. Once selected, this button changes to "Cancel" and "Save" buttons.
	Editable Field	A Field that can be changed.
	Employee	A person working within the company. Employee information is stored as a Contact.
	Field	A single element of information. May be Static or Editable.
	Field Error	An Editable Field displayed in red due to an error in the input.
	GA	An abbreviated term for Google Analytics.
	GA Profile	Created for a website within a user account to track the website's metrics.
	GA Query	A request of data from the profile selected in Google Analytics
	Google Analytics	A utility provided by Google to analyze data gathered from a selected website profile.
	GUI	Graphical User Interface
	Input Error	Any error that may be returned by trying to save an Editable Field with a Field Error.
	List Pane	A searchable, selectable list containing all of the Contacts', Departments', or Report Profiles' names, depending on the selected Tab.
	Maximum Results	The maximum number a returned selected dimension or metric can be.
	Metrics*	The individual measurements of visitor activity on your site. Options include visitors, number of new visits, and the amount of time a user spends on the site.
	Pop-up Window	A window that appears above all of the other GUI windows.
	Preview	Queries GA servers using the selected Report Profile and displays the resulting Report in a Pop-up Window.
	Remove	Removes the selected Contacts from the current Department or the selected Contacts or Departments from a Report. Only visible when "Edit" has been selected for the current Department or Report.

Report	A collection of Dimensions, Metrics, Specified Recipient(s), and other information used in generating a Report.
Report Association	A database table in which each row is a link between a Department or a Contact and a Report.
Report Results	The formatted results returned by GA generated from a query using a specific Report Profile's data.
Revert	Changes settings back to the last saved settings.
Save	Updates the changes made to Editable Fields and stores the new information in the database.
Send	Queries GA servers using the selected Report Profile and emails the Specified Recipient(s) the resulting Report.
Server Error	Any error that may be returned by GA servers.
Settings Pane	A sub-section of the GUI for viewing and editing the preferences.
Specified Recipient(s)	The Contacts or Departments who will be receiving a Report, saved in the Report Association table.
Static Field	A Field that cannot be changed.
Tab	A sub-section of the GUI containing the List Pane and Details Pane for the current selection, either Contacts, Departments, or Report Profiles.

Dimensions*		
Dimension Type	Dimension	Definition
Visitor	visitorType	A boolean indicating if a visitor is new or returning. Possible values: New Visitor, Returning Visitor.
	visitCount	The visit index for a visitor to your website. Each visit from a unique visitor will get its own incremental index starting from 1 for the first visit. Subsequent visits do not change previous visit indices. For example, if a certain visitor has 4 visits to your website, ga:visitCount for that visitor will have 4 distinct values of '1' through '4'.
	daysSinceLastVisit	The number of days elapsed since visitors last visited your website. Used to calculate visitor loyalty.
	userDefinedValue	The value provided when you define custom visitor segments for your website.
Session	visitLength	The length of a visit to your website measured in seconds and reported in second increments. The value returned is a string.
Traffic Sources	referralPath	The path of the referring URL (e.g. document.referrer). If someone places a link to your website on their website, this element contains the path of the page that contains the referring link.
	campaign	When using manual campaign tracking, the value of the utm_campaign campaign tracking parameter. When using AdWords autotagging, the name(s) of the online ad campaign that you use for your website. Otherwise the value (not set) is used.
	source	The source of referrals to your website. When using manual campaign tracking, the value of the utm_source campaign tracking parameter. When using AdWords autotagging, the value is google. Otherwise the domain of the source referring the visitor to your website. (e.g. document.referrer). The value may also contain a port address. If the visitor came to the site without a referrer, the value is (direct)
	medium	The type of referrals to your website. When using manual campaign tracking, the value of the utm_medium campaign tracking parameter. When using AdWords autotagging, the value is ppc. If the user comes from a search engine detected by Google Analytics, the value is organic. If the referrer is not a search engine, the value is referral. If the visitor came directly to the site, and document.referrer is empty, the value is (none).

	keyword	When using manual campaign tracking, the value of the utm_term campaign tracking parameter. When using AdWords autotagging or if a visitor used organic search to reach your website, the keywords used by visitors to reach your website. Otherwise the value is (not set).
	adContent	When using manual campaign tracking, the value of the utm_content campaign tracking parameter. When using AdWords autotagging, the first line of the text for your online Ad campaign. If you are using mad libs for your AdWords content, this field displays the keywords you provided for the mad libs keyword match. Otherwise the value is (not set)
	socialNetwork	Name of the social network. This can be related to the referring social network for traffic sources, or to the social network for social data hub activities. E.g. Google+, Blogger, reddit, etc.
	hasSocialSourceReferral	Indicates visits that arrived to the site from a social source (i.e. A social network such as Google+, Facebook, Twitter, etc.). The possible values are Yes or No where the first letter is capitalized.
Platform/Device	browser	The names of browsers used by visitors to your website. For example, Internet Explorer or Firefox.
	browserVersion	The browser versions used by visitors to your website. For example, 2.0.0.14
	operatingSystem	The operating system used by your visitors. For example, Windows, Linux , Macintosh, iPhone, iPod.
	operatingSystemVersion	The version of the operating system used by your visitors, such as XP for Windows or PPC for Macintosh.
	isMobile	Indicates mobile visitors. The possible values are Yes or No where the first letter must be capitalized.
	mobileDeviceBranding	Mobile manufacturer or branded name (e.g: Samsung, HTC, Verizon, T-Mobile).
	mobileDeviceModel	Mobile device model (e.g.: Nexus S)
	mobileInputSelector	Selector used on the mobile device (e.g.: touchscreen, joystick, clickwheel, stylus).
	mobileDeviceInfo	The branding, model, and marketing name used to identify the mobile device.
Geo/Network	continent	The continents of website visitors, derived from IP addresses.
	subContinent	The sub-continent of website visitors, derived from IP addresses. For example, Polynesia or Northern Europe.
	country	The countries of website visitors, derived from IP addresses.
	region	The region of website visitors, derived from IP addresses. In the U. S., a region is a state, such as New York.
	metro	The Designated Market Area (DMA) from where traffic arrived on your site.
	city	The cities of website visitors, derived from IP addresses.
	latitude	The approximate latitude of the visitor's city. Derived from IP address. Locations north of the equator are represented by positive values and locations south of the equator by negative values.
	longitude	The approximate longitude of the visitor's city. Derived from IP address. Locations east of the meridian are represented by positive values and locations west of the meridian by negative values.
	networkDomain	The domain name of the ISPs used by visitors to your website. This is derived from the domain name registered to the IP address.
	networkLocation	The name of service providers used to reach your website. For example, if most visitors to your website come via the major service providers for cable internet, you will see the names of those cable service providers in this element.
System	flashVersion	The versions of Flash supported by visitors' browsers, including minor versions.
	javaEnabled	Indicates Java support for visitors' browsers. The possible values are Yes or No where the first letter must be capitalized.

	language	The language provided by the HTTP Request for the browser. Values are given as an ISO-639 code (e.g. en-gb for British English).
	screenColors	The color depth of visitors' monitors, as retrieved from the DOM of the visitor's browser. For example 4-bit, 8-bit, 24-bit, or undefined-bit.
	screenResolution	The screen resolution of visitors' monitors, as retrieved from the DOM of the visitor's browser. For example: 1024x738.
Page Tracking	hostname	The hostname from which the tracking request was made.
	pagePath	A page on your website specified by path and/or query parameters. Use in conjunction with ga:hostname to get the full URL of the page.
	pagePathLevel1	This dimension rolls up all the page paths in the first hierarchical level in ga:pagePath.
	pagePathLevel2	This dimension rolls up all the page paths in the second hierarchical level in ga:pagePath.
	pagePathLevel3	This dimension rolls up all the page paths in the third hierarchical level in ga:pagePath.
	pagePathLevel4	This dimension rolls up all the page paths in the fourth hierarchical level in ga:pagePath. All additional levels in the ga:pagePath hierarchy are also rolled up in this dimension.
	pageTitle	The title of a page. Keep in mind that multiple pages might have the same page title.
	landingPagePath	The first page in a user's session, or landing page.
	secondPagePath	The second page in a user's session.
	exitPagePath	The last page in a user's session, or exit page.
	previousPagePath	A page on your website that was visited before another page on your website. Typically used with the ga:nextPagePath dimension.
	nextPagePath	A page on your website that was visited after another page on your website. Typically used with the ga:previousPagePath dimension.
	pageDepth	The number of pages visited by visitors during a session (visit). The value is a histogram that counts pageviews across a range of possible values. In this calculation, all visits will have at least one pageview, and some percentage of visits will have more.
Internal Search	searchUsed	A boolean to distinguish whether internal site search was used in a session. Values are Visits With Site Search and Visits Without Site Search.
	searchKeyword	Search terms used by website visitors on your internal site search.
	searchKeywordRefinement	Subsequent keyword search terms or strings entered by users after a given initial string search.
	searchCategory	The categories used for the internal search if you have this enabled for your profile. For example, you might have product categories such as electronics, furniture, or clothing.
	searchStartPage	A page where the user initiated an internal site search.
	searchDestinationPage	A page that the user visited after performing an internal website search.
Time	date	The date of the visit. An integer in the form YYYYMMDD.
	year	The year of the visit. A four-digit year from 2005 to the current year.
	month	The month of the visit. A two digit integer from 01 to 12.
	week	The week of the visit. A two-digit number from 01 to 53. Each week starts on Sunday.
	day	The day of the month. A two-digit number from 01 to 31.

hour	A two-digit hour of the day ranging from 00-23 in the timezone configured for the account. This value is also corrected for daylight savings time, adhering to all local rules for daylight savings time. If your timezone follows daylight savings time, there will be an apparent bump in the number of visits during the change-over hour (e.g. between 1:00 and 2:00) for the day per year when that hour repeats. A corresponding hour with zero visits will occur at the opposite changeover. (Google Analytics does not track visitor time more precisely than hours.)
nthMonth	Index for each month in the specified date range. Index for the first month in the date range is 0, 1 for the second month, and so on. The index corresponds to ga:month entries.
nthWeek	Index for each week in the specified date range. Index for the first week in the date range is 0, 1 for the second week, and so on. The index corresponds to ga:week entries.
nthDay	Index for each day in the specified date range. Index for the first day (i.e., start-date) in the date range is 0, 1 for the second day, and so on.
dayOfWeek	The day of the week. A one-digit number from 0 (Sunday) to 6 (Monday).

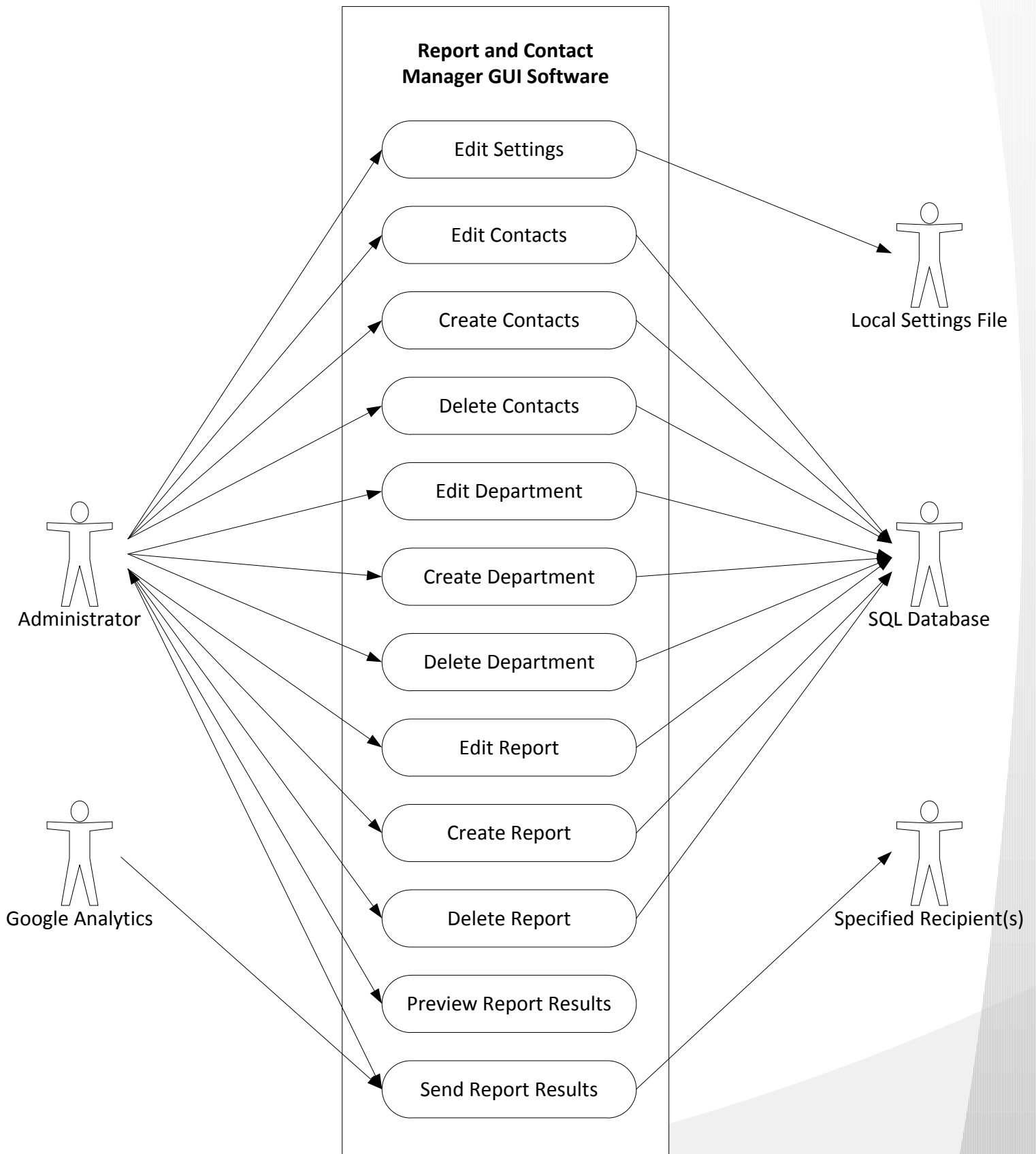
Metrics*		
Metric Type	Metric	Definition
Visitor	visitors	Total number of visitors to your website for the requested time period.
	newVisits	The number of visitors whose visit to your website was marked as a first-time visit.
	percentNewVisits	$(ga:newVisits / ga:visits) * 100$ The percentage of visits by people who had never visited your site before.
Session	visits	Counts the total number of sessions.
	bounces	The total number of single page (or single engagement hit) sessions to your website.
	entranceBounceRate	$(ga:bounces / ga:entrances) * 100$ The percentage of single-page visits (i.e. visits in which the person left your site from the entrance page).
	visitBounceRate	$(ga:bounces / ga:visits) * 100$ The percentage of single-page visits (i.e., visits in which the person left your site from the first page).
	timeOnSite	The total duration of visitor sessions represented in total seconds.
	avgTimeOnSite	$ga:timeOnSite / ga:visits$ The average duration visitor sessions represented in total seconds.
Traffic Sources	organicSearches	The number of organic searches that happened within a session. This metric is search engine agnostic.
Page Tracking	entrances	The number of entrances to your website measured as the first pageview in a session. Typically used with ga:landingPagePath
	entranceRate	$(ga:entrances / ga:pageviews) * 100$ The percentage of pageviews in which this page was the entrance.
	pageviews	The total number of pageviews for your website.
	pageviewsPerVisit	$ga:pageviews / ga:visits$ The average number of pages viewed during a visit to your site. Repeated views of a single page are counted.
	uniquePageviews	The number of different (unique) pages within a session. This takes into both the pagePath and pageTitle to determine uniqueness.

	timeOnPage	How long a visitor spent on a particular page in seconds. Calculated by subtracting the initial view time for a particular page from the initial view time for a subsequent page. Thus, this metric does not apply to exit pages for your website.
	avgTimeOnPage	$\text{ga:timeOnPage} / (\text{ga:pageviews} - \text{ga:exits})$ The average amount of time visitors spent viewing this page or a set of pages.
	exits	The number of exits from your website.
	exitRate	$(\text{ga:exits} / \text{ga:pageviews}) * 100$ The percentage of site exits that occurred out of the total page views.
Internal Search	searchResultViews	The number of times a search result page was viewed after performing a search.
	searchUniques	The total number of unique keywords from internal site search within a session. For example if "shoes" was searched for 3 times in a session, it will be only counted once.
	avgSearchResultViews	$\text{ga:searchResultViews} / \text{ga:searchUniques}$ The average number of times people viewed a search results page after performing a search.
	searchVisits	The total number of sessions that included an internal site search
	percentVisitsWithSearch	$(\text{ga:searchVisits} / \text{ga:visits}) * 100$ The percentage of visits with search.
	searchDepth	The average number of subsequent page views made on your site after a use of your internal search feature.
	avgSearchDepth	$\text{ga:searchDepth} / \text{ga:searchUniques}$ The average number of pages people viewed after performing a search on your site.
	searchRefinements	The total number of times a refinement (transition) occurs between internal site search keywords within a session. For example if the sequence of keywords is: "shoes", "shoes", "pants", "pants", this metric will be one because the transition between "shoes" and "pants" is different.
	searchDuration	The visit duration to your site where a use of your internal search feature occurred.
	avgSearchDuration	$\text{ga:searchDuration} / \text{ga:searchUniques}$ The average amount of time people spent on your site after searching.
	searchExits	The number of exits on your site that occurred following a search result from your internal search feature.
	searchExitRate	$(\text{ga:searchExits} / \text{ga:searchUniques}) * 100$ The percentage of searches that resulted in an immediate exit from your site.
	searchGoal(n) ConversionRate	$(\text{ga:goal}(n)\text{Completions} / \text{ga:searchUniques}) * 100$ The percentage of search visits (i.e., visits that included at least one search) which resulted in a conversion to the requested goal number
	SearchGoalConversionRateAll	$(\text{ga:goalCompletionsAll} / \text{ga:searchUniques}) * 100$ The percentage of search visits (i.e., visits that included at least one search) which resulted in a conversion to at least one of your goals.
	goalValueAllPerSearch	$\text{ga:goalValueAll} / \text{ga:searchUniques}$ The average goal value of a search on your site.
Site Speed	pageLoadTime	Total Page Load Time is the amount of time (in milliseconds) it takes for pages from the sample set to load, from initiation of the pageview (e.g. click on a page link) to load completion in the browser.
	pageLoadSample	The sample set (or count) of pageviews used to calculate the average page load time.
	avgPageLoadTime	$(\text{ga:pageLoadTime} / \text{ga:pageLoadSample}) * 0.001$ The average amount of time (in seconds) it takes for pages from the sample set to load, from initiation of the pageview (e.g. click on a page link) to load completion in the browser.

	domainLookupTime	$(ga:domainLookupTime / ga:speedMetricsSample) * 0.001$ The average amount of time (in seconds) spent in DNS lookup for this page.
	avgDomainLookupTime	$(ga:domainLookupTime / ga:speedMetricsSample) * 0.001$ The average amount of time (in seconds) spent in DNS lookup for this page.
	pageDownloadTime	The total amount of time (in milliseconds) to download this page among all samples.
	avgPageDownloadTime	$(pageDownloadTime / ga:speedMetricsSample) * 0.001$ The average amount of time (in seconds) to download this page.
	redirectionTime	The total amount of time (in milliseconds) spent in redirects before fetching this page among all samples. If there are no redirects, the value for this metric is expected to be 0.
	avgRedirectionTime	$(ga:redirectionTime / ga:speedMetricsSample) * 0.001$ The average amount of time (in seconds) spent in redirects before fetching this page. If there are no redirects, the value for this metric is expected to be 0.
	serverConnectionTime	The total amount of time (in milliseconds) spent in establishing TCP connection for this page among all samples.
	avgServerConnectionTime	$(ga:serverConnectionTime / ga:speedMetricsSample) * 0.001$ The average amount of time (in seconds) spent in establishing TCP connection for this page.
	serverResponseTime	The total amount of time (in milliseconds) your server takes to respond to a user request among all samples, including the network time from user's location to your server.
	avgServerResponseTime	$(ga:serverResponseTime / ga:speedMetricsSample) * 0.001$ The average amount of time (in seconds) your server takes to respond to a user request, including the network time from user's location to your server.
	speedMetricsSample	The sample set (or count) of pageviews used to calculate the averages for site speed metrics. This metric is used in all site speed average calculations including ga:avgDomainLookupTime, ga:avgPageDownloadTime, ga:avgRedirectionTime, ga:avgServerConnectionTime, and ga:avgServerResponseTime.
Exception Tracking	exceptions	The number of exceptions that were sent to Google Analytics.
	fatalExceptions	The number of exceptions where isFatal is set to true.

* Google Analytics Query Explorer. Oct. 18 2012. <http://ga-dev-tools.appspot.com/explorer/>

Use-Case Diagram



Use-Case Descriptions

Brief Description: The "Edit Settings" use case enables the administrator to change or edit the SQL server, email server, and email account being used.

Step-by-Step Description:

1. Administrator selects the "Settings" tab.
2. GUI displays the settings pane.
3. Administrator enters the appropriate information in each field.
4. Administrator clicks "Save Changes" button at the bottom of the GUI.
5. GUI displays the settings pane with the new settings and updates local settings file.

Possible Alternative:

1. Settings file may not be found.
 - a. GUI displays a pop-up window that the settings could not be found.
 - b. GUI displays a blank settings pane.
 - c. Administrator enter settings.
 - d. Administrator clicks "Save Changes" button at the bottom of the GUI.
2. Administrator wishes to cancel changes.
 - a. Administrator clicks "Revert Changes" button.
 - b. GUI displays the settings pane with the previously saved settings.
3. Administrator does not wish to save changes.
 - a. Administrator clicks "No"
 - b. GUI displays settings pane with changed settings, but setting may still be reverted.
4. Administrator may change values and exit with saving changes.
 - a. GUI displays a pop-up window warning that there are unsaved changes.
 - b. Administrator left clicks "Yes".
 - i. Settings are saved.
 - ii. GUI exits.
 - c. Administrator left clicks "No".
 - i. Settings are not saved.
 - ii. GUI exits.

Brief Description:The "Edit Contact" use case enables an administrator to change all or some of an existing contact information.

Step-by-step Description:

1. Administrator selects "Contacts" tab.
2. GUI displays the contacts' list and contacts' details pane screen.
3. Administrator selects contact to be changed from contacts list.
4. GUI displays the selected contact in the contacts' details pane.
5. Administrator left clicks the "Edit" button.
6. GUI updates "Edit" button to "Cancel" and "Save".
7. The fields are changed to editable fields in the contacts' details pane.
8. Administrator changes the fields that need to be updated.

9. Administrator left clicks the "Save" button to keep changes made.
10. GUI displays confirmation message.
11. Administrator left clicks the "Ok" button.
12. The contact's information becomes static again with the changes.

Possible Alternative:

1. Program may fail to connect to database will display database error.
2. Administrator searches for contact's name via search bar instead of choosing from list.
3. Administrator may enter an invalid value into a field.
 - a. The GUI will display a field error.
 - b. Administrator left clicks "Save" button
 - c. The GUI displays that there is an input error.
4. Administrator may changes values and attempt to exit without saving changes.
 - a. GUI displays a pop-up window warning that there are unsaved changes.
 - b. Administrator left clicks "Yes".
 - i. Contact is saved.
 - ii. GUI exits.
 - c. Administrator left clicks "No".
 - i. Contact is not saved.
 - ii. GUI exits.
5. Administrator may change values but cancel all changes.
 - a. Administrator left clicks "Cancel" button.
 - b. The department becomes static again without the changes.

Brief Description:The "Create Contact" use case enables an administrator to add a Contact to the list of existing Contacts.

Step-by-step Description:

1. Administrator selects "Contacts" tab.
2. The GUI displays the contacts' list and contacts' detail pane.
3. Administrator left clicks the create button.
4. The GUI displays an empty editable contact in the contacts' detail pane.
5. Administrator fills in the editable fields with the new contact in appropriate fields.
6. Administrator left clicks "Save" to save new Contact.
7. GUI displays the contacts' list and contacts' detail pane with the newly added contact.

Possible Alternative:

1. Program may fail to connect to database will display database error.
2. Administrator may fail to enter all necessary fields.
 - a. The GUI will display a field error.
 - b. Administrator left clicks "Save" button
 - c. The GUI displays that there is an input error.
3. Administrator may changes values and attempt to exit without saving changes.
 - a. GUI displays a pop-up window warning that there are unsaved changes.
 - b. Administrator left clicks "Yes".
 - i. Contact is saved.
 - ii. GUI exits.

- c. Administrator left clicks "No".
 - i. Contact is not saved.
 - ii. GUI exits.
- 4. Administrator may change values but cancel all changes.
 - a. Administrator left clicks "Cancel" button.
 - b. The department becomes static again without the changes.

Brief Description:The "Delete Contact" use case enables an administrator to delete all or some of the existing Contact(s).

Step-by-step Description:

1. Administrator selects "Contacts" tab.
 2. GUI displays the contacts' list and contacts' details pane.
 3. Administrator selects contact to be deleted from contacts' list.
 4. GUI displays the selected contact in the contacts' details pane.
 5. Administrator left clicks the delete button.
 6. GUI displays a pop-up window asking for confirmation of the deletion.
- NOTE: This will delete contact from both contacts' list and departments' list and is irreversible!
7. Administrator left clicks the "Yes" button to delete contact.
 10. GUI displays updated contacts' list and blank contacts' detail pane.

Possible Alternative:

1. Program may fail to connect to database will display database error.
2. Administrator searches for contact's name via search bar instead of choosing from list.
3. Administrator may cancel deletion.
 - a. Administrator left clicks the "Cancel" button.
 - b. GUI displays the contact in the contacts' details pane.
4. Administrator exits before clicking "Cancel" or "Yes" buttons.
 - a. Contacts is not deleted and exit is not completed.

Brief Description:The "Edit Department" use case enables an administrator to change all or some of an existing departments' information.

Step-by-step Description:

1. Administrator selects "Departments" tab.
2. GUI displays the departments' list and departments' details pane.
3. Administrator selects department to be changed from departments' list.
4. GUI displays the selected department in the departments' details pane.
5. Administrator left clicks the "Edit" button.
6. GUI updates "Edit" button to "Cancel" and "Save".
7. The fields are changed to editable fields in the departments' details pane.
8. Administrator changes the fields that need to be updated.

9. Administrator left clicks the "Save" button to keep changes made.
10. GUI displays a confirmation message.
11. Administrator left clicks "Ok" button.
12. GUI displays the department with static fields in the departments' detail pane.

Possible Alternative:

1. Program may fail to connect to database will display database error.
2. Administrator searches for department by name via search bar instead of choosing from list.
3. Administrator wishes to add contacts to department.
 - a. Administrator left clicks "Add" button while editing the department.
 - b. A pop-up window opens in the GUI that displays the contacts' list.
 - c. Administrator selects contact(s) from contacts' list.
 - d. Administrator left clicks "Ok".
 - e. The pop-up window closes and GUI displays the editable department screen with contact(s) added to contacts field for the selected department.
4. Administrator wishes to remove contacts from department.
 - a. Administrator selects contact to be removed from department's contacts' list while editing the department.
 - b. Administrator left clicks "Remove" button.
 - c. GUI displays the the editable department in the departments' detail pane with the contact removed from the department's contacts' list.
5. Administrator may fail to enter all necessary fields.
 - a. The GUI will display a field error.
 - b. Administrator left clicks "Save" button
 - c. The GUI displays that there is an input error.
6. Administrator may change values and attempt to exit without saving changes.
 - a. GUI displays a pop-up window warning that there are unsaved changes.
 - b. Administrator left clicks "Yes".
 - i. Department is saved.
 - ii. GUI exits.
 - c. Administrator left clicks "No".
 - i. Department is not saved.
 - ii. GUI exits.
7. Administrator may change values but cancel all changes.
 - a. Administrator left clicks "Cancel" button.
 - b. The department becomes static again without the changes

Brief Description:The "Create Department" use case enables an administrator to add a department to the list existing departments

Step-by-step Description:

1. Administrator selects "Departments" tab.
2. GUI displays the departments' list and departments' details pane.
3. Administrator left clicks the create button.
4. GUI displays a empty editable department in the departments' detail pane.
5. Administrator fills in department's information in correct fields.
6. Administrator left clicks add button.

7. A popup window opens up in the GUI with the contacts' list.
8. Administrator selects contact(s) to be added from contacts' list.
9. The popup window closes and the GUI displays the department with editable fields and department's contacts' list is updated with newly added contact(s)
- NOTE: Department must have at least one Contact to be saved.
10. Administrator left clicks "Save" button to create the department.
11. GUI displays the departments' list with the newly created department and departments' detail pane with the newly created department with static fields.

Possible Alternative:

1. Program may fail to connect to database will display database error.
2. Administrator searches for department by name via search bar instead of choosing from list.
3. Administrator may fail to enter all necessary fields.
 - a. The GUI will display a field error.
 - b. Administrator left clicks "Save" button
 - c. The GUI displays that there is an input error.
4. Administrator may changes values and attempt to exit without saving changes.
 - a. GUI displays a pop-up window warning that there are unsaved changes.
 - b. Administrator left clicks "Yes".
 - i. Department is saved.
 - ii. GUI exits.
 - c. Administrator left clicks "No".
 - i. Department is not saved.
 - ii. GUI exits.
5. Administrator may change values but cancel all changes.
 - a. Administrator left the "Cancel" button.
 - b. The GUI displays the departments' list and blank departments' detail pane.

Brief Description:The "Delete Department" use case enables an administrator to delete department from the list of existing departments

Step-by-step Description:

1. Administrator selects "Departments" tab
2. GUI displays the departments' list and departments' details pane.
3. Administrator selects department to be deleted from the departments' list.
4. GUI displays the selected department in the departments' details pane.
5. Administrator left clicks delete button.
6. GUI displays a pop-up window asking for confirmation of the deletion.
7. Administrator left clicks "Yes" button to delete department.
- NOTE: Contacts will be removed from department, but will not be deleted. The deletion of a department is irreversible.
8. GUI displays the departments' list without the deleted department and a blank departments' details pane.

Possible Alternative:

1. Program may fail to connect to database will display database error.
2. Administrator searches for department by name via search bar instead of choosing from list.

3. Administrator may changes values and attempt to exit without saving changes.
 - a. Department is not deleted and exit is completed.
4. Administrator may want to cancel the delete.
 - a. Administrator left clicks the "Cancel" button.
 - b. The GUI will display the departments' list and departments' details pane with the department selected.

Brief Description:The "Edit Report" use case documents the user flow required to edit an existing report.

Step-by-step Description:

1. Administrator selects "Reports" Tab.
2. GUI displays the reports' list and the reports' details pane.
3. Administrator selects the desired report from the reports' List.
4. GUI displays the selected report in the reports' details pane.
5. Administrator left clicks the "Edit" button.
6. GUI updates "Edit" button to "Cancel" and "Save".
7. The fields are changed to editable fields in the reports' details pane.
8. Administrator changes the fields that need to be updated.
9. Administrator left clicks the "Save" button to keep changes made.
10. GUI displays confirmation message.
11. Administrator left clicks the "Ok" button.
12. The report's fields becomes static again with the changes.

Possible Alternative:

1. Program may fail to connect to database will display database error.
2. Administrator searches for report by name via search bar instead of choosing from list.
3. Administrator wishes to add recipients.
 - a. Administrator left clicks "Add" button next to recipients field while editing the report.
 - b. A pop-up window opens in the GUI that displays the contacts' and departments' lists.
 - c. Administrator selects contact(s) or department(s) from contacts' list.
 - d. Administrator left clicks "Ok".
 - e. The pop-up window closes and GUI displays the editable report screen with recipients added to recipients field for the selected report.
4. Administrator wishes to remove recipients.
 - a. Administrator selects contact(s) or department(s) to be removed from report's recipients field while editing the report.
 - b. Administrator left clicks "Remove" button.
 - c. GUI displays the the editable report in the reports' detail pane with the contact(s) or department(s) removed from the recipients field.
5. Administrator may fail to enter all necessary fields.
 - a. The GUI will display a field error.
 - b. Administrator left clicks "Save" button
 - c. The GUI displays that there is an input error.
6. Administrator may change values and attempt to exit without saving changes.
 - a. GUI displays a pop-up window warning that there are unsaved changes.
 - b. Administrator left clicks "Yes".

- i. Report is saved.
 - ii. GUI exits.
- c. Administrator left clicks "No".
 - i. Report is not saved.
 - ii. GUI exits.
- 7. Administrator may change values but cancel all changes.
 - a. Administrator left clicks "Cancel" button.
 - b. The department becomes static again without the changes

Brief Description: The "Create Report" use case enables an administrator to create a new report and save the report to Reports.

Step-by-step Description:

1. Administrator selects "Reports" tab.
 2. GUI displays the reports' list and the reports' details pane
 4. Administrator left clicks create button.
 5. GUI displays an empty report with editable fields in the reports' detail pane.
 6. Administrator inputs the report profile in the appropriate editable fields.
- NOTE: Reports require at least one metric.
- NOTE: Reports are not required to have specified recipients. They may be added when sending report results.
7. Administrator left clicks "Save" button.
- NOTE: Reports are saved in database.
8. GUI displays a confirmation message.
 9. Administrator left clicks "Ok".
 10. GUI displays the reports' list with the newly added report and reports' detail pane with the newly added report profile with static fields.

Possible Alternative:

1. Program may fail to connect to database will display database error.
2. Administrator wishes to add recipients.
 - a. Administrator left clicks "Add" button next to the recipients field.
 - b. A pop-up window opens in the GUI that displays the contacts' and departments' lists.
 - c. Administrator selects contact(s) or department(s) from contacts' list.
 - d. Administrator left clicks "Ok".
 - e. The pop-up window closes and GUI displays the editable report screen with recipients added to recipients field for the new report.
3. Administrator may fail to enter all necessary fields.
 - a. The GUI displays a field error.
 - b. When the administrator left clicks the "Save" button, the GUI displays that there is a input error.
4. Administrator may change values and attempt to exit without saving changes.
 - a. GUI displays a pop-up window warning that there are unsaved changes.
 - b. Administrator left clicks "Yes".
 - i. Report is saved.
 - ii. GUI exits.

- c. Administrator left clicks "No".
 - i. Report is not saved.
 - ii. GUI exits.
- 5. Administrator may change values but cancel.
 - a. Administrator left clicks the "Cancel" button.
 - b. The GUI displays the reports' list without the new report and a blank reports' detail pane.

Brief Description:The "Delete Report" use case enables an administrator to delete a report.

Step-by-step Description:

1. Administrator selects "Reports" tab
2. GUI displays the reports' list and a blank reports' details pane.
3. Administrator selects report to be deleted from the reports' list.
4. Administrator left clicks delete button.
5. GUI displays a pop-up window asking for confirmation of the deletion.
6. Administrator left clicks "Yes" button.
- NOTE: Deleting a report is irreversible.
7. The GUI displays the reports' list and a blank reports' detail pane.

Possible Alternative:

1. Program may fail to connect to database will display database error.
2. Administrator may changes values and attempt to exit without saving changes.
 - a. Report is not deleted and exit is completed.
3. Administrator want to cancel the delete.
 - a. Administrator left clicks "Cancel" button.
 - b. The GUI displays the reports' list with the selected report and report in the reports' detail pane.

Brief Description:The "Preview Report Results" use case documents the user flow required to generate but not send report results.

Step-by-step Description:

1. Administrator selects "Reports" Tab.
2. The GUI displays the reports' list and the reports' details pane.
3. Administrator selects the desired report from the reports' list.
4. GUI displays the selected report in the reports' details pane.
5. Administrator left clicks the "Preview" button.
6. GUI displays a pop-up window containing the report results.
7. Administrator left clicks "Ok" to close the pop-up window.

Possible Alternative:

1. Program may fail to connect to database will display database error.
2. Program may fail to connect to GA server and will display server error.
3. Administrator searches for report by name via search bar instead of choosing from list.

Brief Description:The "Send Report Results" use case enables an administrator to send report results to specified recipients.

Step-by-step Description:

1. Administrator selects "Reports" Tab.
2. The GUI displays the reports' list and the reports' details pane.
3. Administrator selects the desired report from the reports' list.
4. GUI displays the selected report in the reports' details pane.
5. Administrator left clicks the "Send" button.
6. GUI displays a pop-up window containing the recipients pre-populated from the report.
7. Administrator left clicks "Send Report" button.
8. GUI closes pop-up window and sends report results to all specified recipients.
9. GUI display a pop-up window confirming that the report results have been sent.

Possible Alternative:

1. Program may fail to connect to database will display database error.
2. Program may fail to connect to GA server and will display server error.
3. Program may fail to connect to mail server and will display server error.
4. Administrator wishes to add recipients.
 - a. Administrator left clicks the "Send" button.
 - b. GUI displays a pop-up window containing the recipients pre-populated from the report.
 - c. Administrator left clicks the "Add" button.
 - d. Another pop-up window opens in the GUI that displays the contacts' list and departments' list.
 - e. Administrator selects contact(s) and department(s) from the lists.
 - f. Administrator left clicks "Ok".
 - g. The second pop-up window closes and GUI displays the updated recipients.
5. Administrator attempts to send report results with no specified recipients.
 - a. Administrator left clicks the "Send" button.
 - b. GUI displays a message that there are no specified recipients with the option to add recipients.
 - c. Administrator may add recipients if desired by clicking the "Add" button.
6. Administrator does not wish to send report results.
 - a. Administrator left clicks the "Send" button.
 - b. GUI displays a pop-up window containing the recipients pre-populated from the report.
 - c. Administrator left clicks the "Cancel" button.

Use-Case Scenarios

1. Edit Settings

- a. Administrator selects "Settings" tab.
- b. Administrator types "mail.ku.edu" into the text field beside "Email Server:".
- c. Administrator types "suzyq" into the text field beside "Email Username:".
- d. Administrator types "ccr" into the text field beside "Email Password:".
- e. Administrator then left clicks "Save Changes".
- f. Updated settings for the email server are saved.

2. Edit an existing Contact.

- a. Administrator selects "Contact" tab.
- b. Administrator clicks in the search bar (next to the magnifying glass).
- c. Administrator enters "Cornish" (This is the last name).
- d. A list appears displaying all users with "Cornish" in any field (ex: name, phone number, email, etc.).
- e. Administrator selects the entry with the name "Conrad Cornish".
- f. The right side of the window will be populated with "Conrad Cornish" 's contact information.
- g. Administrator clicks the "Edit" button.
- h. Administrator changes the telephone from "911" to "1-900-656-1224".
- i. Administrator also changes the birthday from "01/01/1900" to "01/01/1970".
- j. Administrator changes email from ccornish@miners.com to Cornishriotinto.com
- k. Administrator clicks "Save".
- l. A window appears explaining that the email field is incorrect.
- m. Administrator clicks "Ok".
- n. Administrator changes the email field to Cornish@riotinto.com.
- o. Administrator clicks "Save".
- p. A window appears confirming the save.

3. Create a new Contact.

- a. Administrator selects "Contacts" tab.
- b. Administrator clicks the create ("+") button.
- c. Administrator types in the contact information for the employee being created into the appropriate fields.
 - First Name: Jesse
 - Last Name: Pinkman
 - Telephone: 1-505-654-3434
 - Email: jpinkman@vamanos.com

- Birthday: 05/15/1985
 - d. Administrator clicks "Save" to save new employee.
 - e. A new record with the above information is created in the database.
 - f. A dialog appears confirming the addition.
4. Delete an existing Contact.
- a. Administrator selects "Contacts" tab.
 - b. Administrator clicks the search bar (next to the magnifying glass at the top).
 - c. Administrator enters "Cornish" into the search bar.
 - d. Administrator selects the entry with the name "Conrad Cornish".
 - e. The right side of the window is populated with "Conrad Cornish" 's contact information.
 - f. Administrator clicks delete button (the trash bin).
 - g. A dialog appears to confirm the deletion.
 - h. Administrator clicks "Yes".
 - i. The record is now deleted from the database.
 - j. All fields are now empty.
5. Edit an existing Department.
- a. Administrator selects "Departments" tab.
 - b. Administrator selects "Accounting" from the list of possible departments.
 - c. The contact information for that department and the employees in that department are populated into the correct fields.
 - d. Administrator clicks the "Edit" button.
 - e. Administrator clicks "Add" button
 - f. Administrator selects the name "Hank Schrader" from the list of employees in the window that appears.
 - g. Administrator clicks "Ok".
 - h. The name "Hank Schrader" now appears in the list of employees in the department.
 - i. Administrator clicks the "Save" button.
 - j. The changes are saved to the database.
 - k. Administrator is returned to the edit department screen.
6. Create a new Department.
- a. Administrator selects "Departments" tab.
 - b. Administrator clicks create ("+") button.
 - c. Administrator enters the following information into the department being created.
 - Department Name: Accounting
 - Department Telephone: 1-888-555-5555
 - Department Email: acct@vamanos.com
 - d. Administrator clicks "Add" button.

- e. Administrator searches for the name "Walter White" in the window that appears and selects the name returned.
 - f. Administrator clicks "Ok" button to add employee to department's employee list.
 - g. The list is now updated with the name "Walter White".
 - h. Administrator clicks "Save" button to create the department and is returned to list of existing departments.
7. Delete an existing Department.
- a. Administrator selects "Departments" tab
 - b. Administrator selects "Accounting" from the list of departments.
 - c. The department contact information and list of employees in the department are retrieved from the database and shown in the right side of the window.
 - d. Administrator clicks delete button (the trash bin).
 - e. A dialog appears asking the user to confirm the deletion.
 - f. Administrator clicks "Yes" button to delete department and returns to new list of existing departments.
 - g. The department and its associations are deleted from the database.
 - h. Administrator is returned to the "Departments" screen.
8. Edit an existing Report.
- a. Administrator selects "Reports" tab.
 - b. Administrator select the "Weekly TPS" from the list of reports.
 - c. Administrator clicks "Edit" button.
 - d. Administrator changes Start Date to "10/14/2012".
 - e. Administrator changes End Date to "10/02/2012".
 - f. Administrator clicks "Save" button.
 - g. A dialog appears informing the administrator that the end date is invalid.
 - h. Administrator changes End Date to "10/20/2012".
9. Create a new Report.
- a. Administrator selects "Reports" tab
 - b. Administrator clicks create ("+") button.
 - c. Administrator enters the following information into the report being created.
 - Start Date: 01/01/2012
 - End Date: 06/01/2012
 - Dimensions: Operating System
 - Metric: Unique Visits
 - The Sort and Filter fields are left empty.
 - d. Administrator clicks "Save" button.
 - e. A new record with the above information is created in the database.

- f. A dialog appears confirming the addition.

10. Delete an existing Report.

- a. Administrator selects "Reports" tab
- b. Administrator selects "Daily Mobile Visitors" report from the list of reports.
- c. Administrator clicks delete (the trash bin).
- d. A dialog appears asking for confirmation of the deletion.
- e. Administrator clicks "Yes".
- f. The report is deleted from the database.
- g. Administrator is returned to the "Reports" screen.

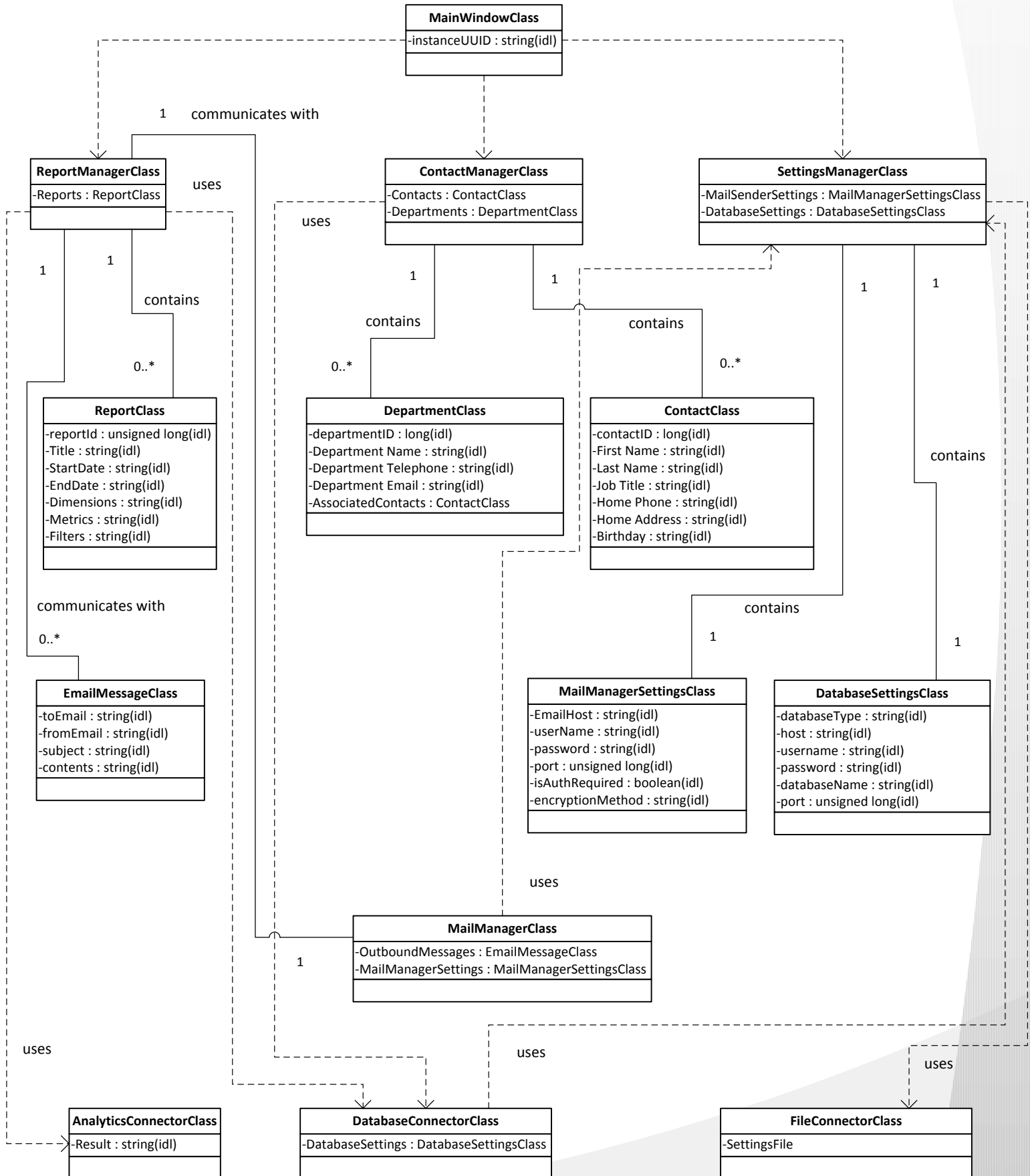
11. Preview Report Results

- a. Administrator selects "Reports" tab.
- b. Administrator selects the "Weekly IE Visitors" report from the list of reports.
- c. The right side of the window populates with the "Weekly IE Visitors" report information.
- d. Administrator clicks "Preview" button.
- e. A pop-up window showing the formatted results for the "Weekly IE Visitors" report is displayed.
- f. Administrator clicks "Ok" to close the preview window.

12. Send Report Results.

- a. Administrator selects "Reports" tab.
- b. Administrator selects the "Weekly Report for IT" report from the list of reports.
- c. The right side of the window populates with the "Weekly Report for IT" report information.
- d. Administrator clicks "Send" button.
- e. A dialog appears asking for confirmation and shows the specified recipients (IT). An option to add additional recipients is included.
- f. Administrator clicks "Add".
- g. A dialog appears with the list of contacts and departments.
- h. Administrator selects "John Doe" and clicks "Ok".
- i. "John Doe" is now added to the recipients.
- j. Administrator clicks "Send Report".
- k. The report results are generated and sent to the IT department and John Doe.
- l. A dialog appears confirming that the message was successfully sent.

Class Diagram



Class-Responsibility-Collaboration (CRC) Cards

Class: MainWindowClass
Responsibility: <ol 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: <ol style="list-style-type: none">1. ContactManagerClass(Dependency)2. ReportManagerClass(Dependency)3. SettingsManagerClass (Dependency)

Class: ReportManagerClass
Responsibility: <ol style="list-style-type: none">1. Check with MainWindowClass if database is ready.2. Send request for report definitions to DatabaseConnectorClass.3. Receive report definitions from DatabaseConnectorClass.4. Make a new ReportClass for each possible report definition.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 instances into EmailMessageClass instances.15. Push EmailMessageClass instances to MailManagerClass.
Collaboration: <ol style="list-style-type: none">1. AnalyticsConnectorClass(Dependency)2. DatabaseConnectorClass(Dependency)3. ContactManagerClass4. EmailMessageClass5. MailManagerClass6. MainWindowClass7. ReportClass

Class: ContactManagerClass
Responsibility: <ol style="list-style-type: none"> 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: <ol style="list-style-type: none"> 1. DatabaseConnectorClass(Dependency) 2. ContactClass 3. DepartmentClass

Class: SettingsManagerClass
Responsibility: <ol style="list-style-type: none"> 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: <ol style="list-style-type: none"> 1. FileConnectorClass (Dependency) 2. DatabaseSettingsClass 3. MailManagerSettingsClass 4. MainWindowClass

Class: ReportClass
Responsibility: <ol style="list-style-type: none"> 1. Respond to changes in the report definition 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: <ol style="list-style-type: none"> 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 exists.
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: <ol style="list-style-type: none"> 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**

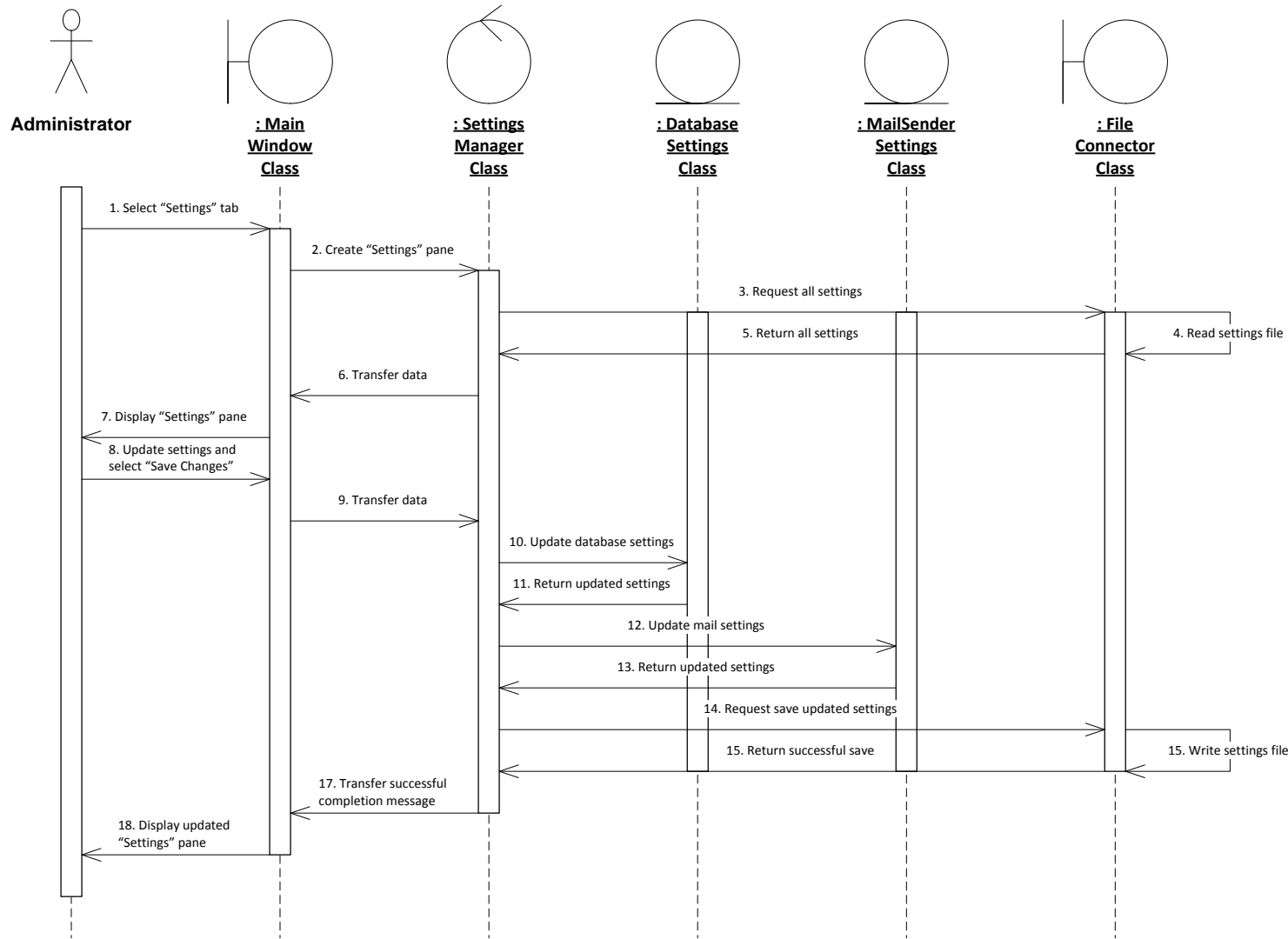
Responsibility:

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.

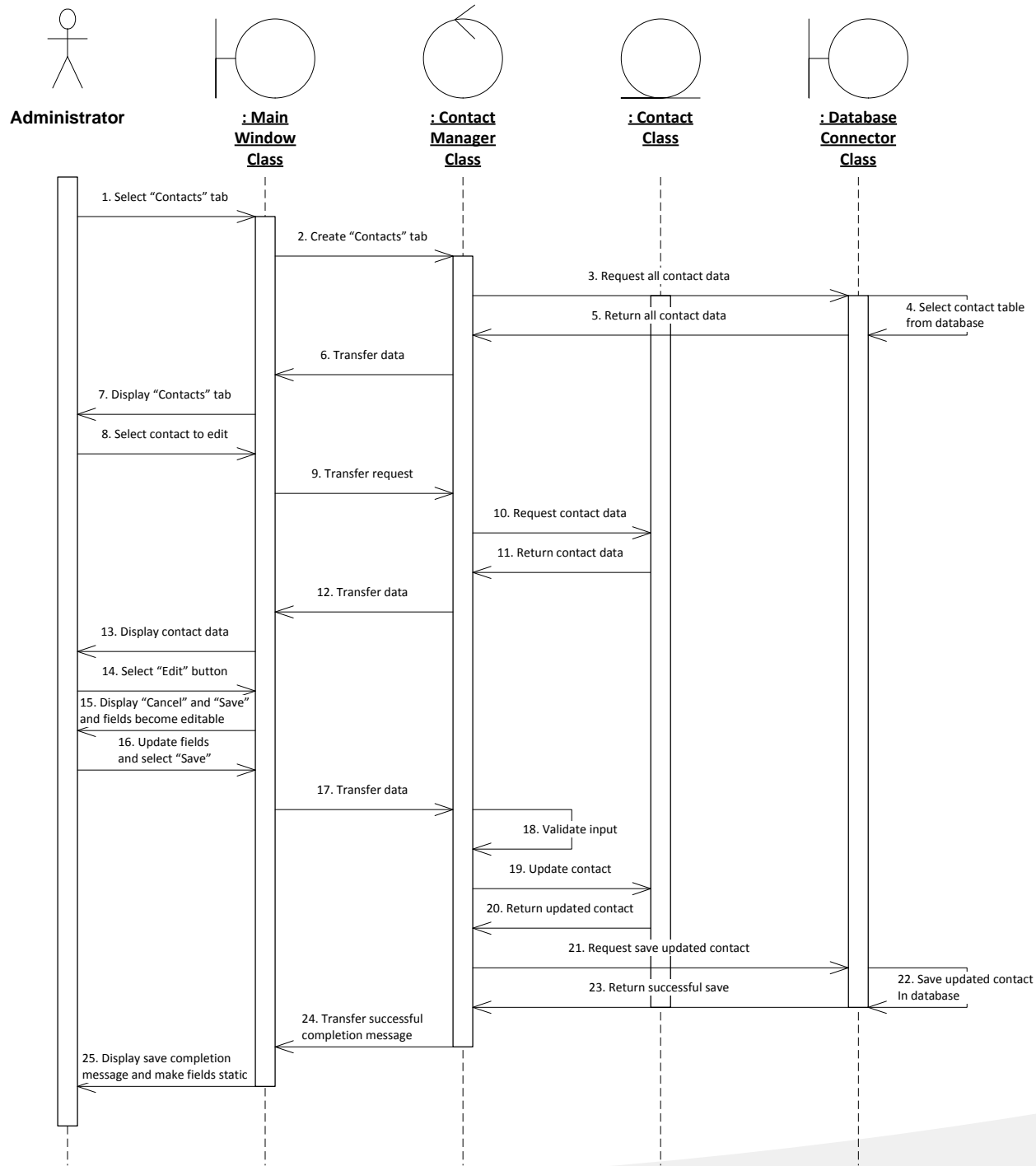
Collaboration:

1. **SettingsManagerClass**

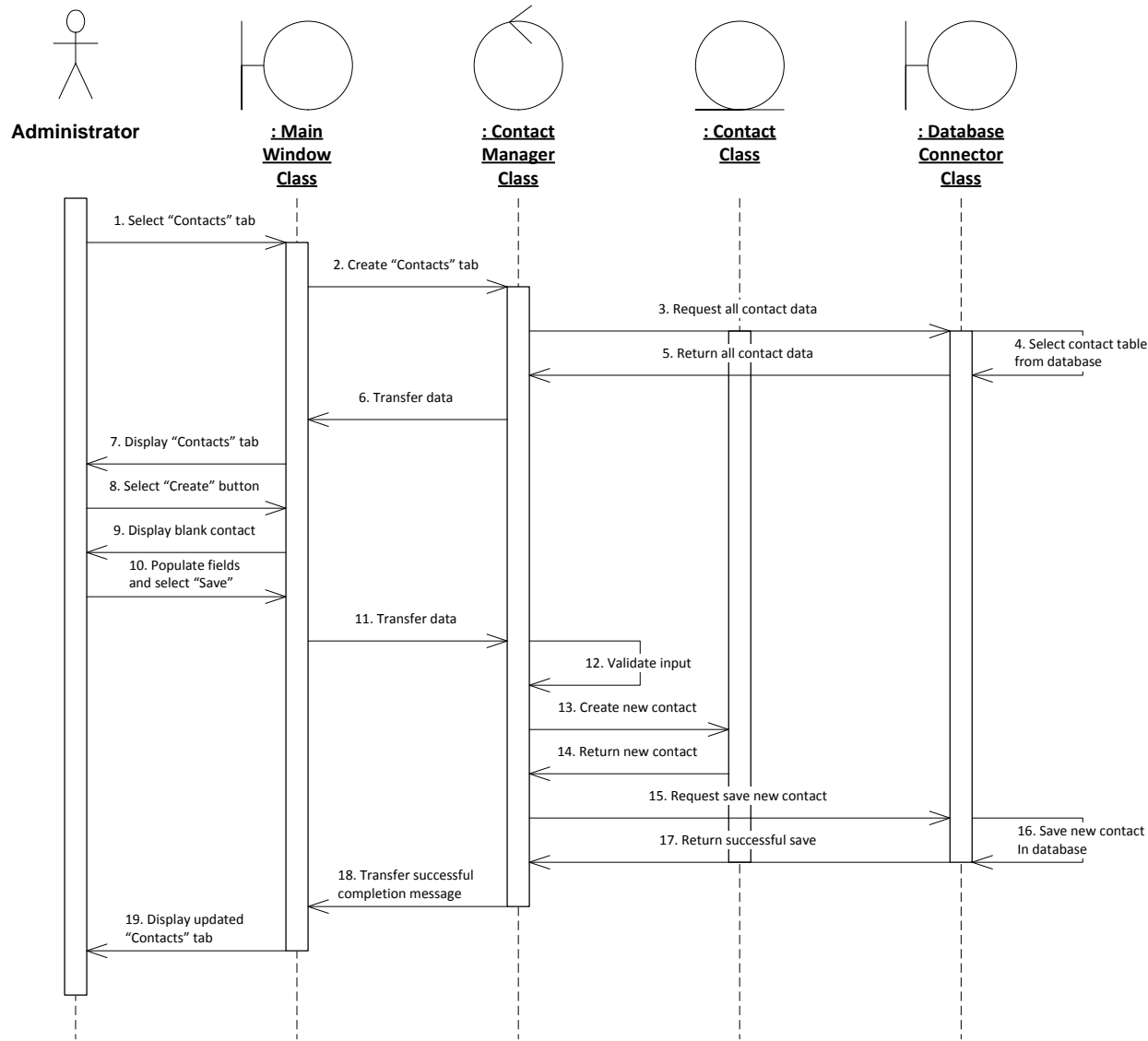
Edit Settings



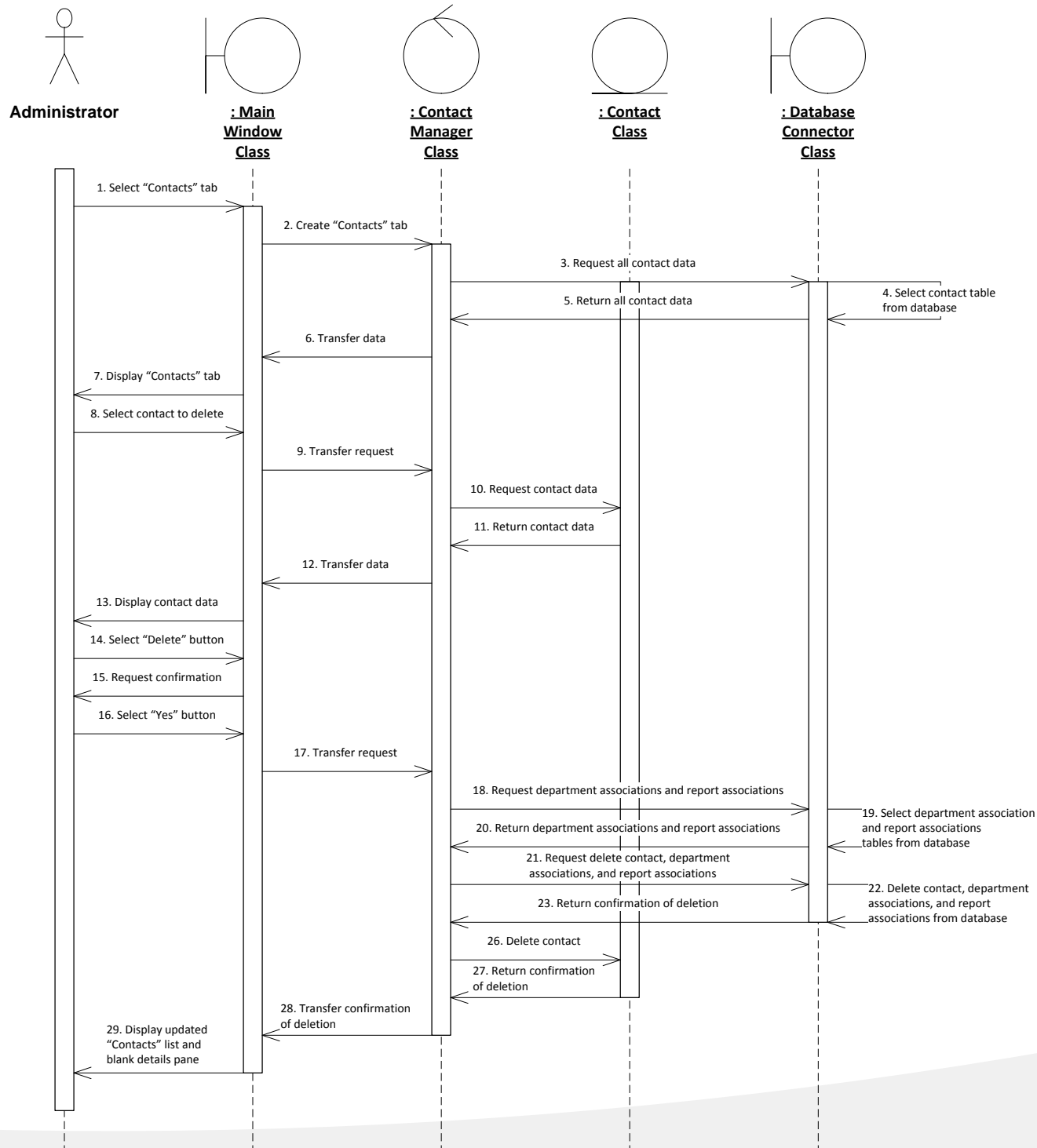
Edit Contact



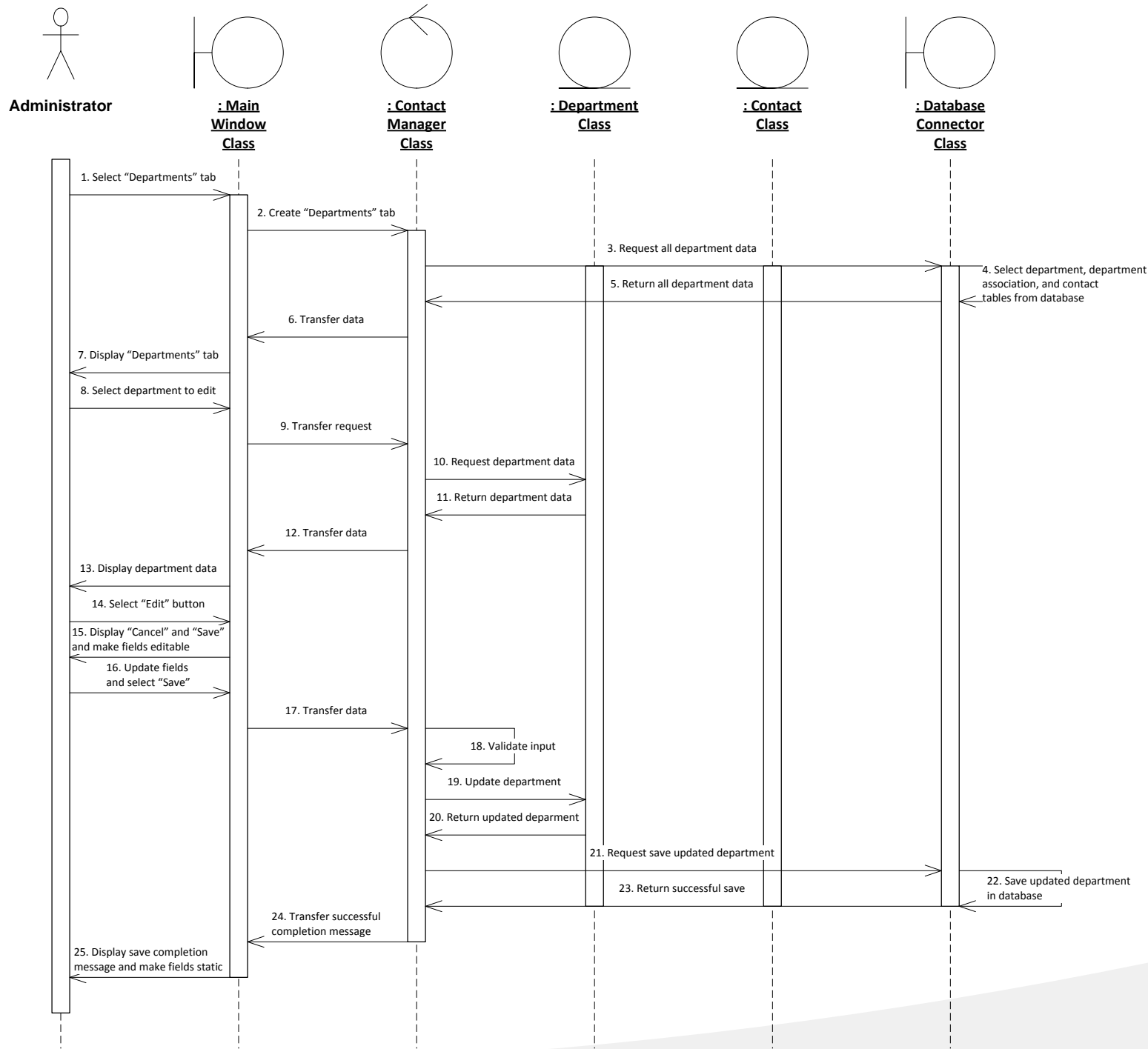
Create Contact



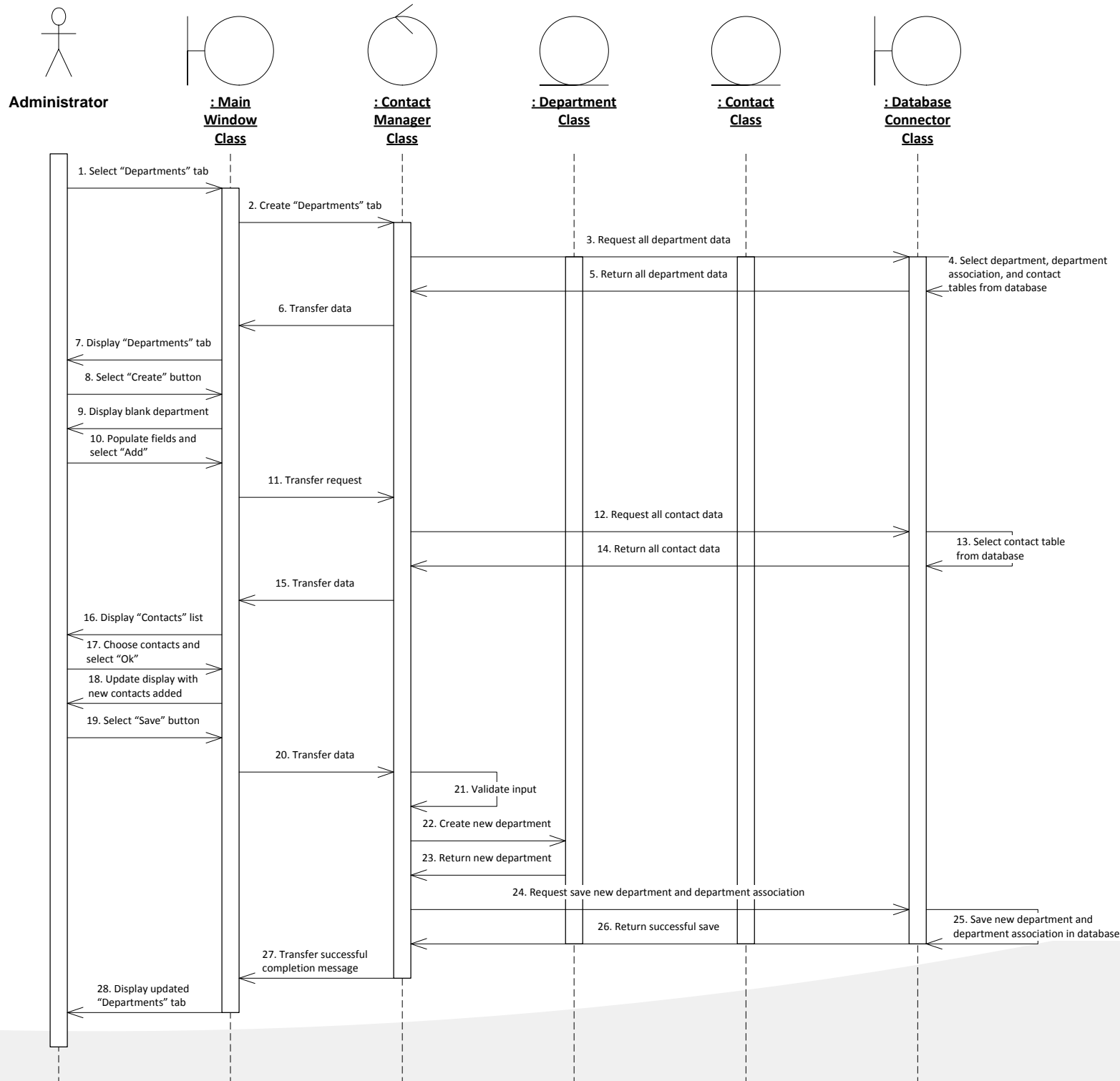
Delete Contact



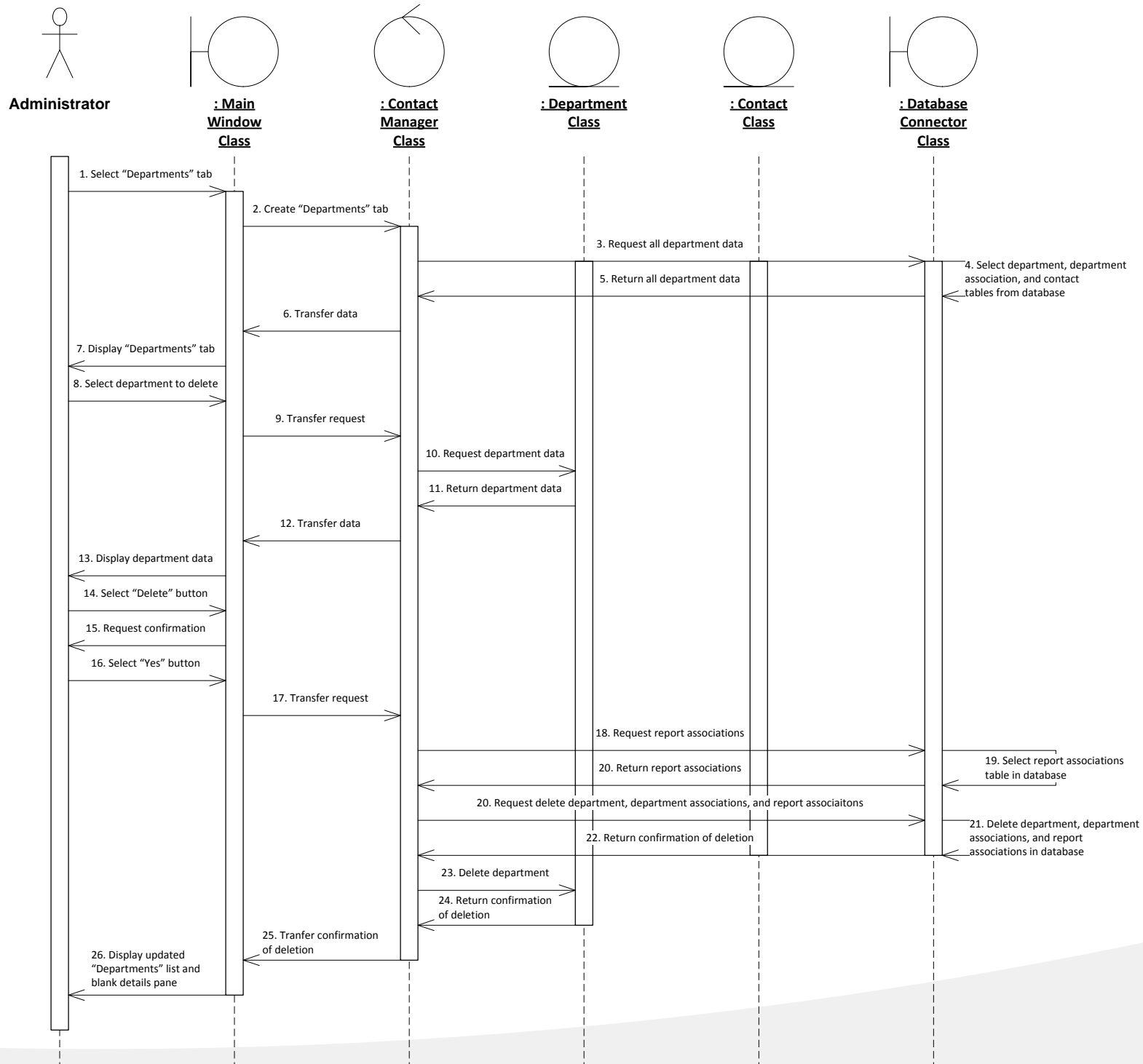
Edit Department



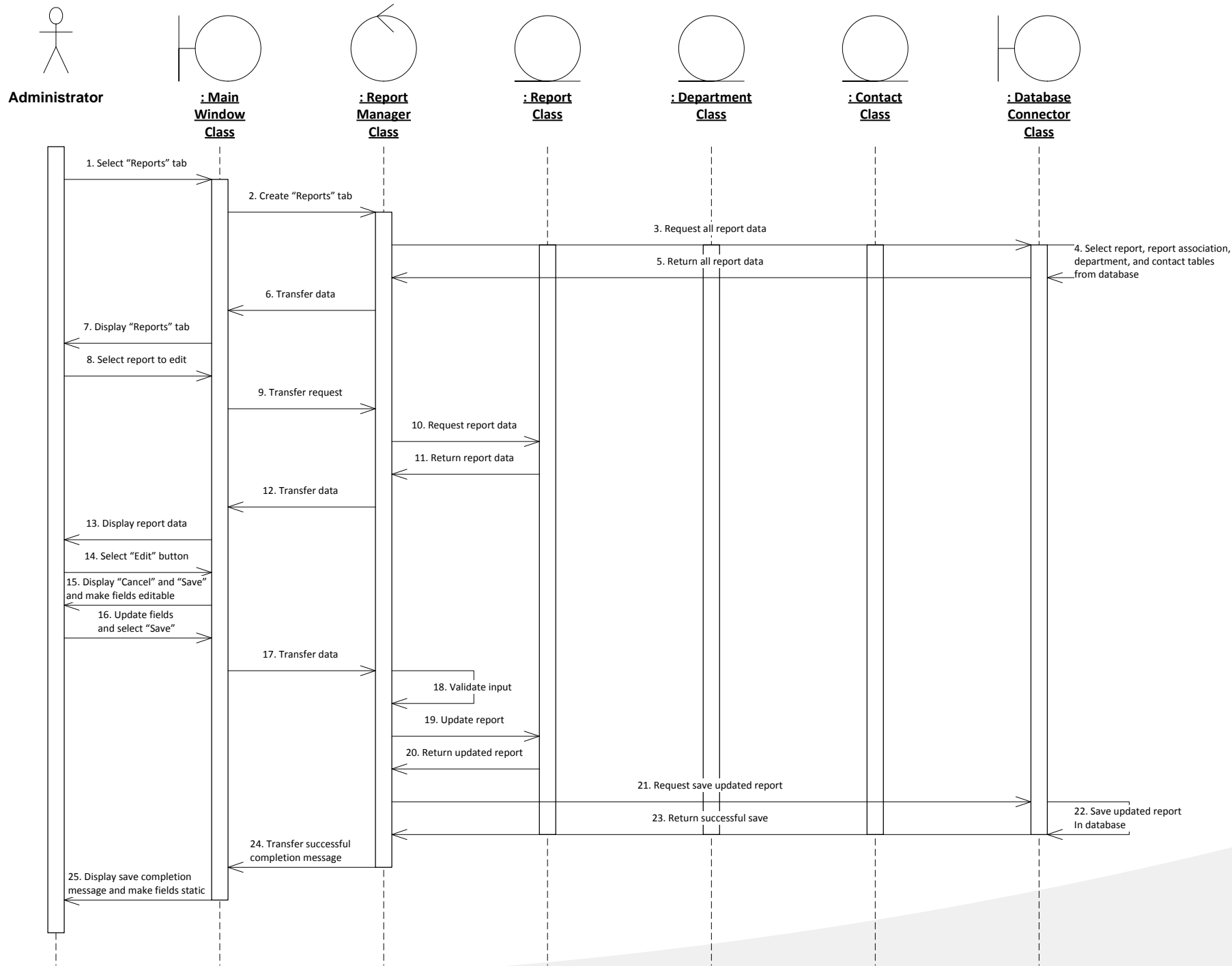
Create Department



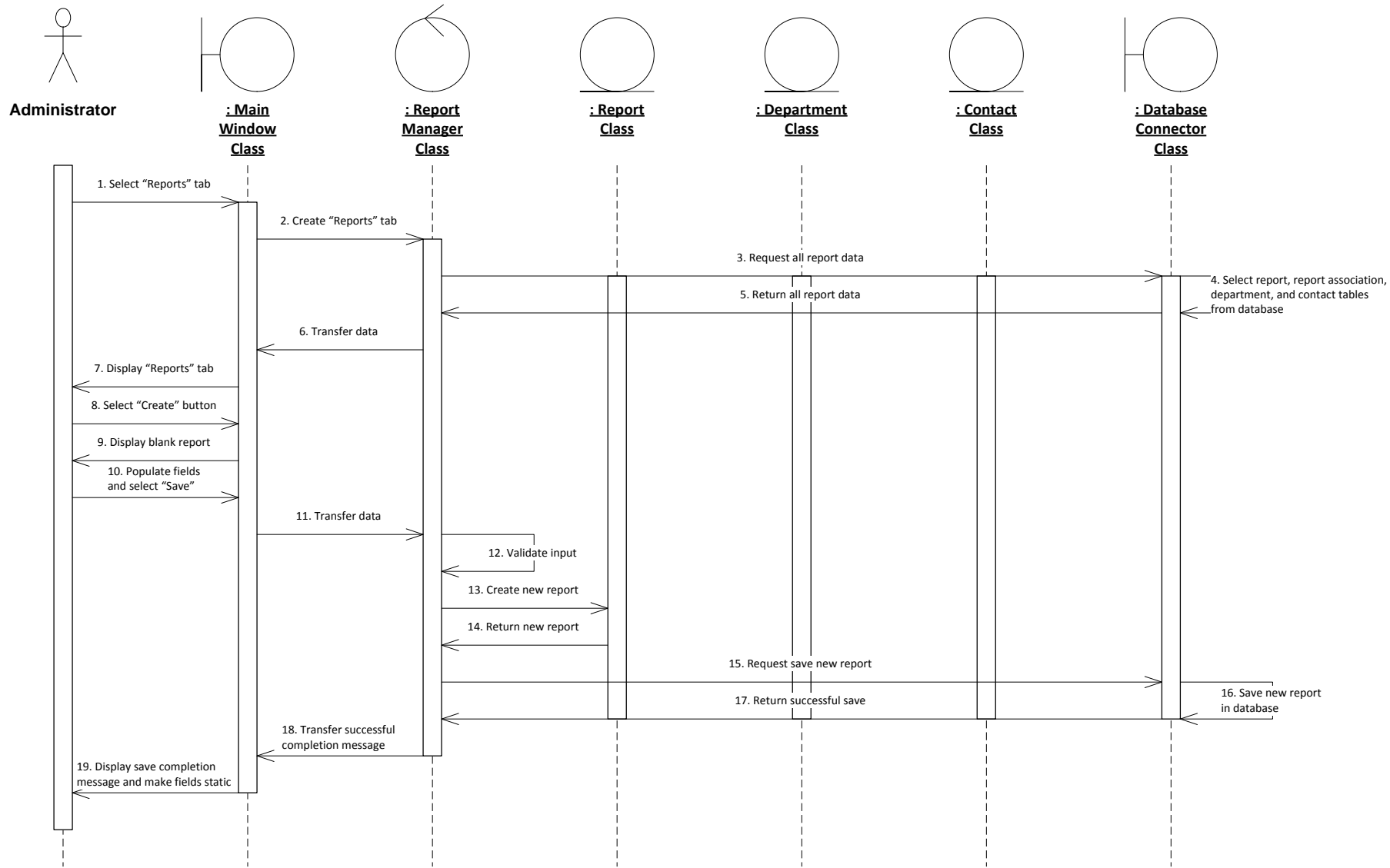
Delete Department



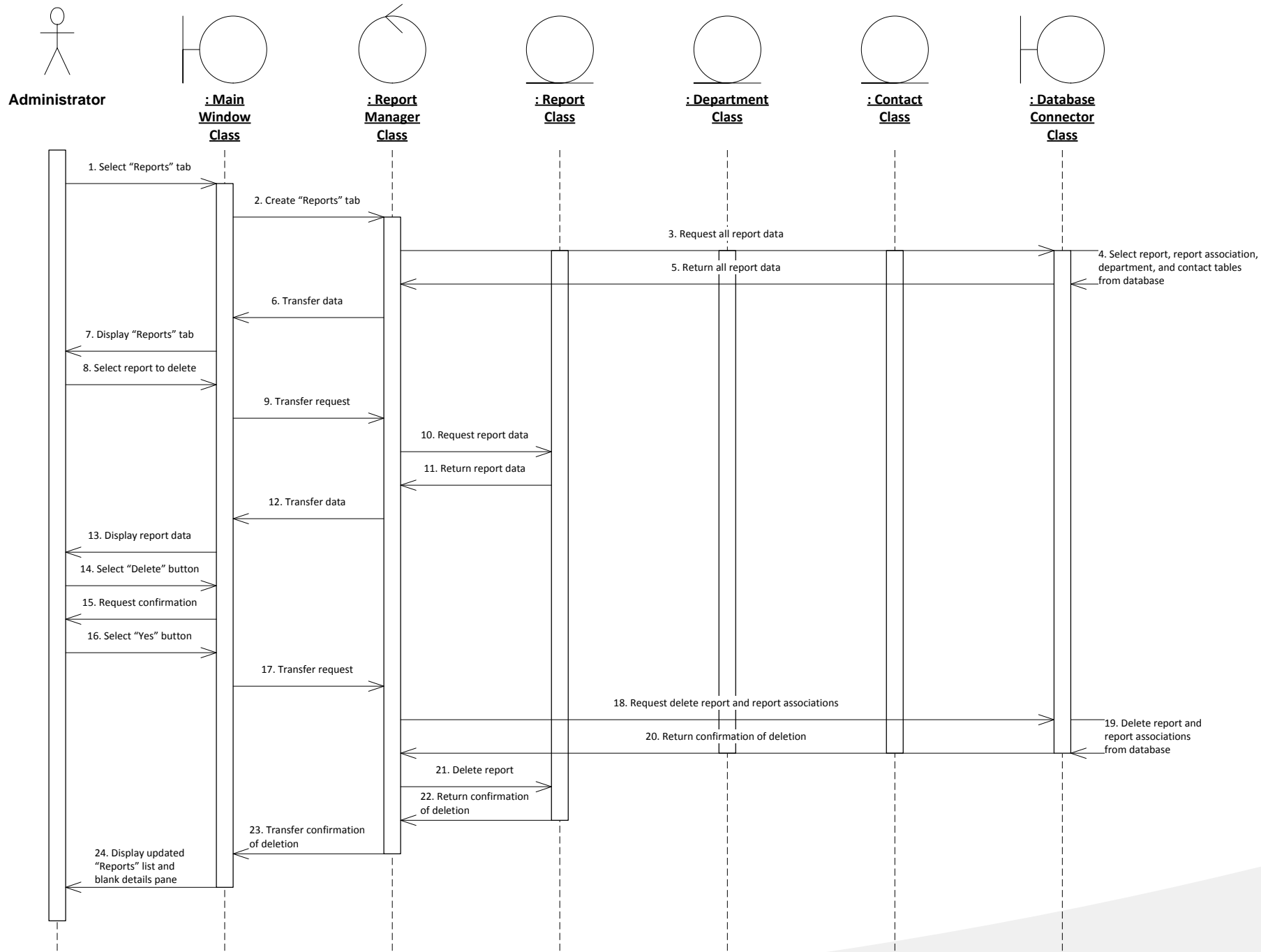
Edit Report



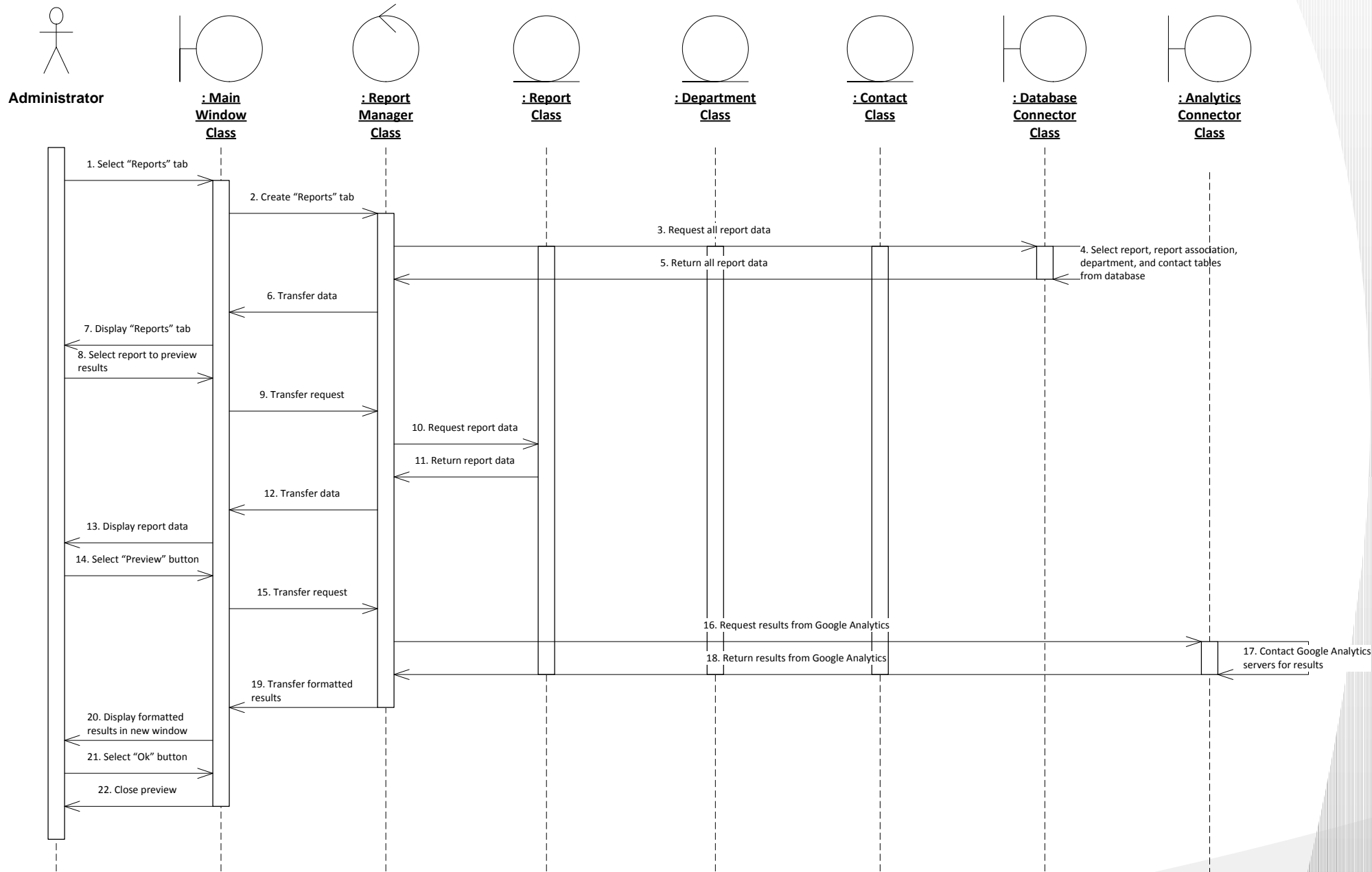
Create Report



Delete Report



Preview Report Results



Send Report Results

