Project 2, Part 1: Wolf Travel Tours ▶

Project 2, Part 1: Wolf Travel Tours

Project 2 requires you to go through standard software development phases to design, implement, and test a complete Java program consisting of multiple source code files and JUnit test cases.

Deliverables and Deadlines

The project comes in two parts. Deliverables for Part 1 are:

1. Design document that includes a UML class diagram
2. Black box test plan

Part 1 Due Date: Friday, March 22, 2019 at 11:45 PM

Late Deadline (Design): Saturday, March 23, 2019 at 9:00 AM

Late Deadline (BBTP): Sunday, March 24, 2019 at 11:45 PM

Reminder

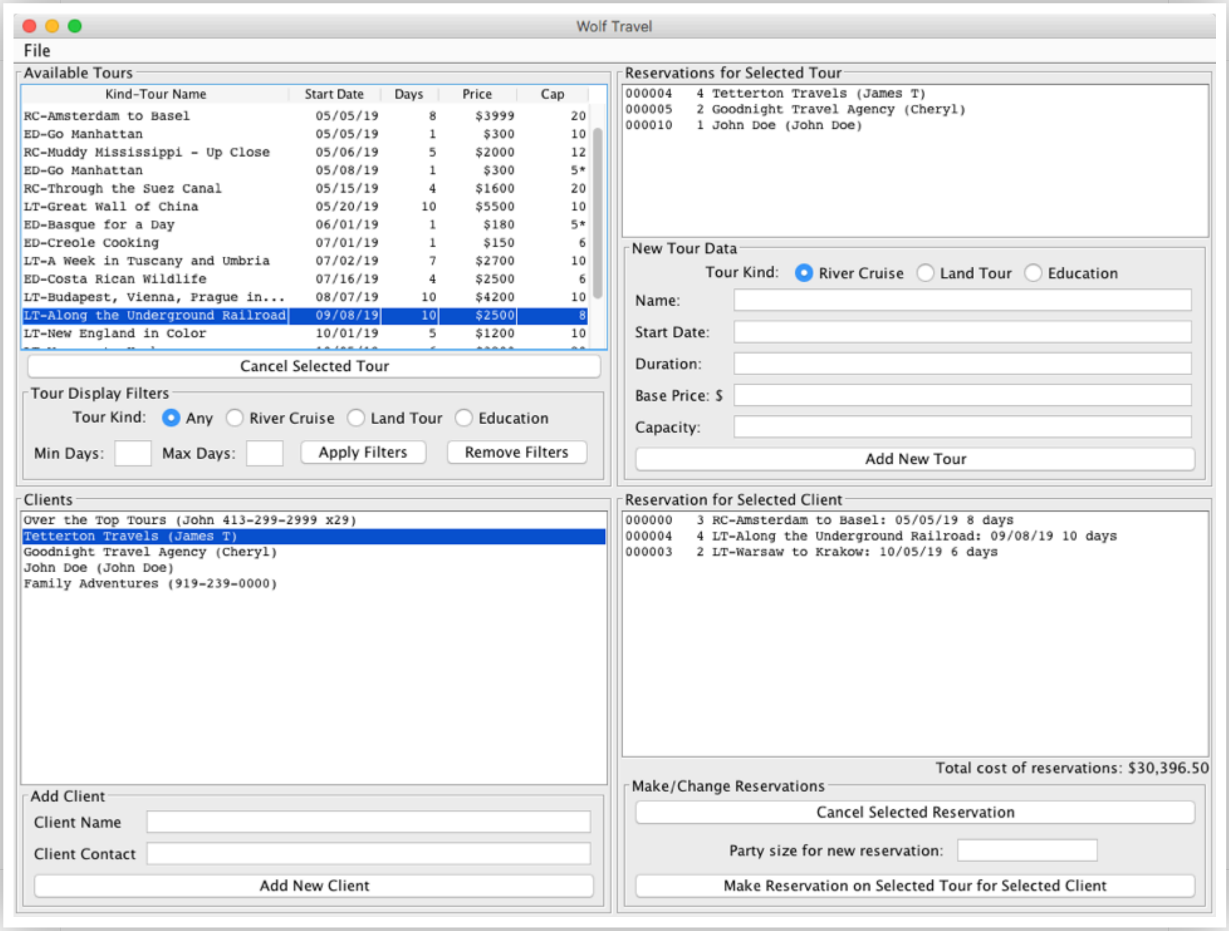
You will *not* be working with a partner for Part 1 of *any* project. All work must be strictly your own.

Problem Overview

Your boss just put in a bid to develop software to help Wolf Travel Company manage its business. Wolf Travel specializes in three kinds of tours: river cruises, land tours, and educational trips/workshops. Wolf Travel clients include not just individuals but also groups and other travel agencies. The purpose of the proposed software is to keep track of Wolf Travel's clients, the tours it offers, and the reservations that clients make for the different tours. You are tasked with writing a prototype to enhance your boss's bid.

Your fine colleague Buster has already written a graphical user interface (GUI) for the prototype as shown in Figure 1. The top half of the GUI is devoted to tours, with the list of available tours on the left and the reservations for the currently selected tour on the right. The bottom half of the GUI is devoted to clients, where the list of clients is on the left and the reservations for the currently selected client is on the right.

The GUI has a File menu for loading and saving files, for creating new files, and for quitting. Buttons and text fields enable the user to add new tours, cancel tours, add new clients, add new reservations, and cancel reservations.



Wolf Travel
File
Available Tours
Cancel Selected Tour
Tour Display Filters
Tour Kind: [✓ Any] River Cruise Land Tour Education
Min Days:
Max Days:
Apply Filters
Remove Filters
Reservations for Selected Tour
New Tour Data
Tour Kind: [✓ River Cruise] Land Tour Education
Name:
Start Date:
Duration:
Base Price: $
Capacity:
Add New Tour
Clients
Add Client
Client Name
Client Contact
Add New Client
Reservation for Selected Client
Make/Change Reservations
Cancel Selected Reservation
Party size for new reservation:
Make Reservation on Selected Tour for Selected Client

Kind-Tour Name	Start Date	Days	Price	Cap
RC-Amsterdam to Basel	05/05/19	8	$3999	20
ED-Go Manhattan	05/05/19	1	$300	10
RC-Muddy Mississippi - Up Close	05/06/19	5	$2000	12
ED-Go Manhattan	05/08/19	1	$300	5*
RC-Through the Suez Canal	05/15/19	4	$1600	20
LT-Great Wall of China	05/20/19	10	$5500	10
ED-Basque for a Day	06/01/19	1	$180	5*
ED-Creole Cooking	07/01/19	1	$150	6
LT-A Week in Tuscany and Umbria	07/02/19	7	$2700	10
ED-Costa Rican Wildlife	07/16/19	4	$2500	6
LT-Budapest, Vienna, Prague in...	08/07/19	10	$4200	10
LT-Along the Underground Railroad	09/08/19	10	$2500	8
LT-New England in Color	10/01/19	5	$1200	10

```
000004   4 Tetterton Travels (James T)
000005   2 Goodnight Travel Agency (Cheryl)
000010   1 John Doe (John Doe)
```

```
Over the Top Tours (John 413-299-2999 x29)
Tetterton Travels (James T)
Goodnight Travel Agency (Cheryl)
John Doe (John Doe)
Family Adventures (919-239-0000)
```

```
000000   3 RC-Amsterdam to Basel: 05/05/19 8 days
000004   4 LT-Along the Underground Railroad: 09/08/19 10 days
000003   2 LT-Warsaw to Krakow: 10/05/19 6 days
```

Total cost of reservations: $30,396.50

Figure 1: Wolf Travel GUI displaying tours, clients, and reservations.

Wolf Travel has already established conventions for clients, tours, and reservations, which are described here.

Each client has a name and a contact. The *Clients* list on the bottom left of Figure 1 shows the client name first, then the contact in parentheses. Client names must begin with alphabetic characters and client contacts must start with alphabetic characters or digits. Two clients are considered the same if they have the same name and the same contact, ignoring case.

Each tour has multiple pieces of information:

1. Tour kind. The list in *Available Tours* at the top left in Figure 1 shows the kind first, with RC representing river cruise, LT to representing land tour, and ED representing educational trip.
2. Tour name. Cannot be blank, all whitespace, or contain the colon character (':'). The name of the first tour in the *Available Tours* list in Figure 1 is "Amsterdam to Basel."
3. Start date. An actual date in the form mm/dd/yy.
4. Duration in days. A whole number larger than 0.
5. Base price in whole dollars. The cost of a reservation for a tour depends on the base price, the number of people in the reservation party, and the kind of tour.
6. Capacity. The maximum number of people the tour can accommodate. This number is fixed except for educational trips. The capacity of an educational trip will double exactly once as needed to accommodate a new reservation, at which time its capacity becomes fixed. Capacities of other kinds of tours are automatically always fixed.

Two tours are considered the same if they have the same name (ignoring case), the same start date, and the same duration.

A reservation can be for parties of various sizes but only a single client and a single tour. It is possible to make a reservation for a client as long as the tour has enough spaces left to accommodate everyone in the party. Each reservation has four pieces of information:

1. A confirmation code, which is a 6-digit number/string in the range 000000 through 999999.
2. A tour.
3. A client.
4. The number of people in the party.

Requirements

The following use cases describe the interactions between the proposed Wolf Travel software system and its users. For the prototype you must create, there is only one actual user, the person who manages Wolf Travel tours.

Use Case 1: Startup

Preconditions: None.

Main Flow: The user starts the program [S1][S2]. The user can create a new Wolf Travel data file [UC4] or load an existing one [UC2].

Subflows:

[S1] The program starts, displaying the Wolf Travel user interface, which is similar to the one shown in Figure 1 except all buttons are disabled and no data are displayed.
The GUI has a menu to create a *New* data file, *Load* data from an existing file, *Save* current data, and *Exit* to quit. The menu choice *File -> Save* is initially disabled.
The main window of the GUI has major panels in each of its four quadrants:
Upper left quadrant: *Available Tours* to show tours and cancel tours along with *Tour Display Filters* to filter the *Available Tours* according to kind and duration.
Upper right quadrant: *Reservations for Selected Tour* to display all reservations for the selected tour along with *New Tour Data* to add new tours.
Lower left quadrant: *Clients* to show all clients and *Add Client* to add new clients.
Lower right quadrant: *Reservations for Selected Client* to show all reservations for the currently selected client and *Make/Change Reservations* for adding or canceling reservations.

[S2] The user optionally opens an existing Wolf Travel data file by choosing *File -> Load* [UC2]. All buttons are then enabled.
-or-
[S3] The user optionally starts a new file by choosing *File -> New* [UC4]. All buttons are then enabled.

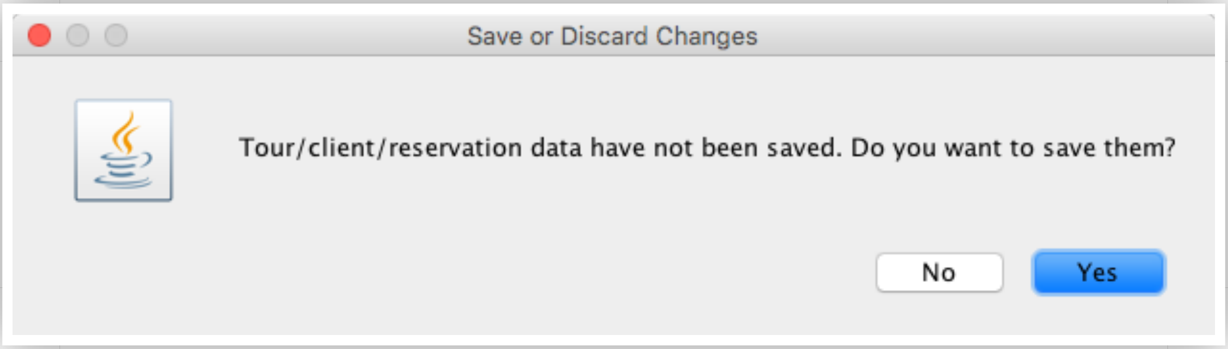
Use Case 2: Load a Wolf Travel data file

Preconditions: The program is running [UC1], and the user interface is open.

Main Flow: The user selects *File -> Load* from menu bar then browses for and selects the desired data file from a File Open dialog [S1][S2]. The system opens and reads the file [S3], then populates the user interface with the file data [S3][S4].

Subflows:

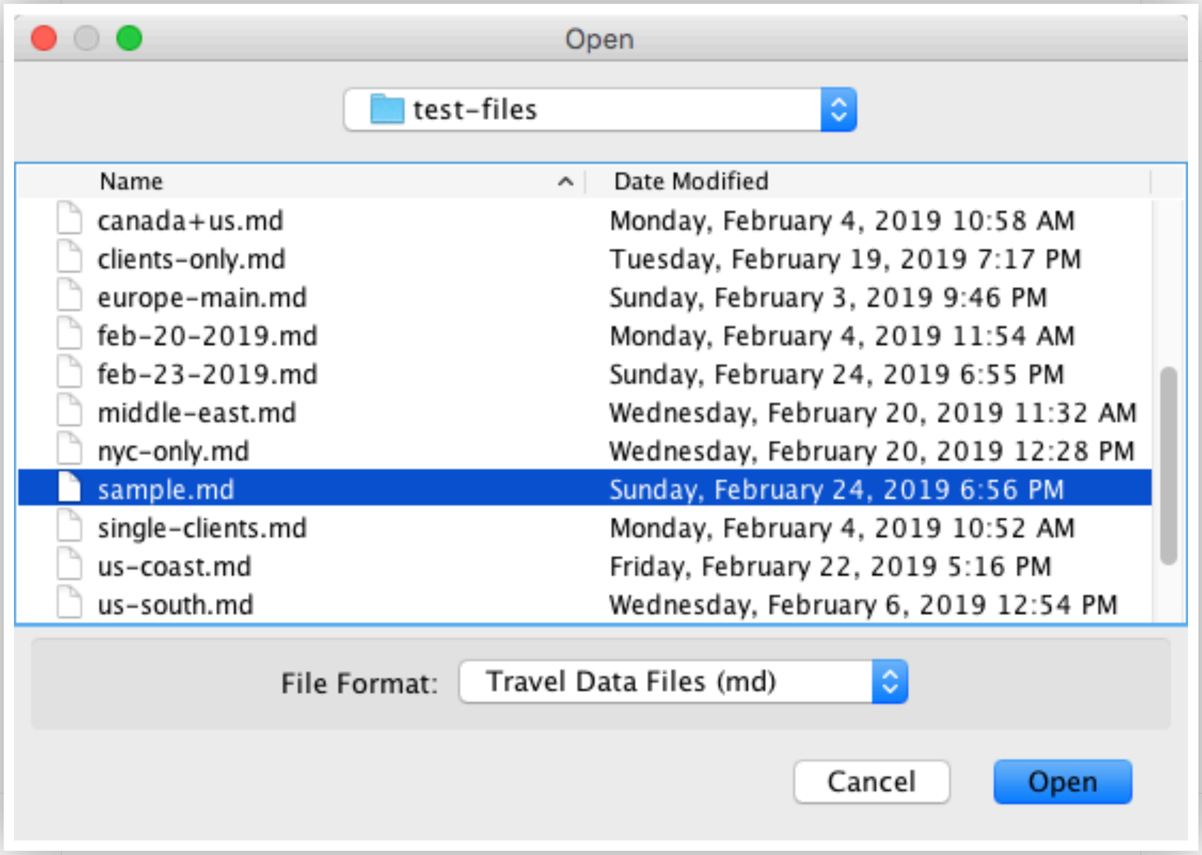
[S1] If the system has unsaved data, the user is first prompted to save it as shown in Figure 2. If the user clicks **No** in the dialog, no data are saved. If the user clicks **Yes** instead, save operation proceeds [UC3] prior to the next step.



Tour/client/reservation data have not been saved. Do you want to save them?

Figure 2: Dialog to save unsaved data.

[S2] A dialog opens as shown in Figure 3, and the user browses for and selects the desired data file [E1].



Name	Date Modified
canada+us.md	Monday, February 4, 2019 10:58 AM
clients-only.md	Tuesday, February 19, 2019 7:17 PM
europe-main.md	Sunday, February 3, 2019 9:46 PM
feb-20-2019.md	Monday, February 4, 2019 11:54 AM
feb-23-2019.md	Sunday, February 24, 2019 6:55 PM
middle-east.md	Wednesday, February 20, 2019 11:32 AM
nyc-only.md	Wednesday, February 20, 2019 12:28 PM
sample.md	Sunday, February 24, 2019 6:56 PM
single-clients.md	Monday, February 4, 2019 10:52 AM
us-coast.md	Friday, February 22, 2019 5:16 PM
us-south.md	Wednesday, February 6, 2019 12:54 PM

File Format: Travel Data Files (md)

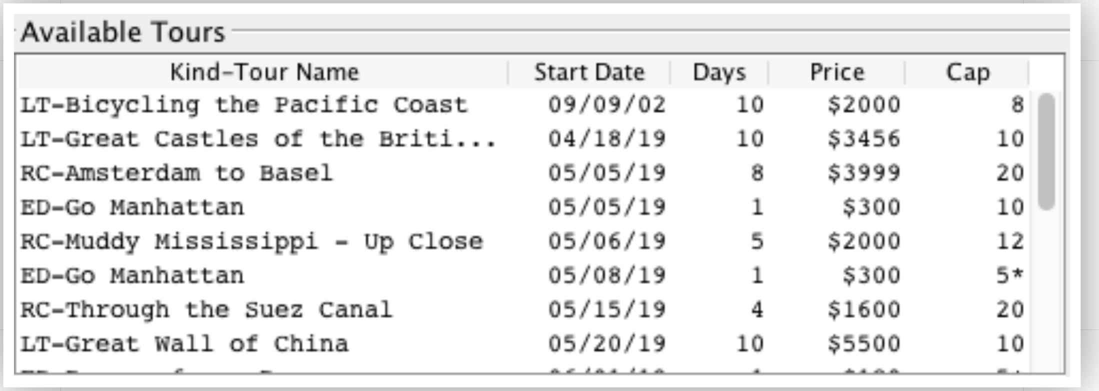
Figure 3: Dialog to open a file.

[S3] The system opens and reads the data file [E2].
A valid data file has extension *.md* and it may have embedded blank lines or extra whitespace at the beginning or end of lines. Each valid file is structured according to the following:

Lines with client information come before any lines with tour or reservation information. Information for each client is on a line by itself. The first part of each such line is the client's name, which must begin with an alphabetic character. The last part of each such line is the client's contact information inside parenthesis. For example:
Circle Tour Agency (John at 919-919-9191)
A line with tour information must precede any line with reservation information. Every line with tour information begins with optional whitespace then the character #, followed by the tour type (RC for river cruise, LT for land tour, ED for educational tour) and a dash and the tour name and a colon, then the start date in the form mm/dd/yy, then the duration of the tour, $ and the base price, and finally the capacity. If it is still possible to increase the capacity, an asterisk comes immediately after the capacity

number. Here are two examples of valid tour lines:
#LT-Basel to Rotterdam: 12/09/19 6 $2500 10
#ED-A Day at MOMA: 03/14/2020 1 $175 8*
Each line with reservation information follows the tour that the reservation applies to. Each such line begins with a 6-digit confirmation code, then the number of people in the party, and finally the client name and the client contact in parentheses. For example:
000005 2 Circle Tour Agency (John at 919-919-9191)

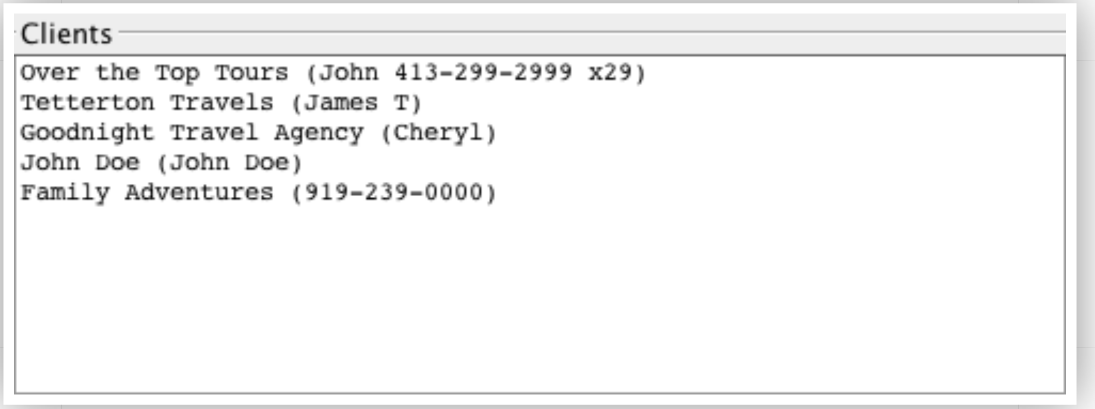
[S4] The system populates the *Available Tours* area with the list of all tours. Tours are listed in ascending order according to their start dates. If two tours have the same start date, those two are ordered according to their names (where the comparison is made without regard to case). If two tours have the same start date and the same name (ignoring case), they are ordered by their durations. No single tour is initially selected. Figure 4 illustrates the tour display format.


Available Tours

Kind-Tour Name	Start Date	Days	Price	Cap
LT-Bicycling the Pacific Coast	09/09/02	10	$2000	8
LT-Great Castles of the Briti...	04/18/19	10	$3456	10
RC-Amsterdam to Basel	05/05/19	8	$3999	20
ED-Go Manhattan	05/05/19	1	$300	10
RC-Muddy Mississippi - Up Close	05/06/19	5	$2000	12
ED-Go Manhattan	05/08/19	1	$300	5*
RC-Through the Suez Canal	05/15/19	4	$1600	20
LT-Great Wall of China	05/20/19	10	$5500	10

Figure 4: Available Tour list.

[S5] The system populates the *Clients* area with the list of all clients. Clients are listed in the order in which they were read. (Clients who are subsequently added will be listed below the ones read from the file.) No single client is initially selected. Figure 5 illustrates the client display format.


Clients

```
Over the Top Tours (John 413-299-2999 x29)
Tetterton Travels (James T)
Goodnight Travel Agency (Cheryl)
John Doe (John Doe)
Family Adventures (919-239-0000)
```

Figure 5: Client list.

The Wolf Travel data file sample.md was used to populate the system with data shown in Figure 1.

Alternative Flows:

[E1] If the use clicks **Cancel** ionn the file Open dialog, no file is opened and the open file process aborts.
[E2] If the selected file is not a valid data file or an error occurs while reading the selected file, an error dialog opens with message "Error opening file." When the user clicks **OK,** the dialog closes, and the load file process aborts. Errors can come from multiple sources, including but not limited to the following:

Lines out of order (such as reservations or clients coming after trips)
Any error in data format (such as illegal dates format or values).
Any error in data validity (such as missing/blank contact information for a client or tour duration that's not at least 1 day)
Reservations without clients listed earlier in the file
Lines not formatted as described above (such a tour without a leading # or having a name that is not followed by :)
The same tour occurring more than once in the data file
The same client occurring more than once in the data file

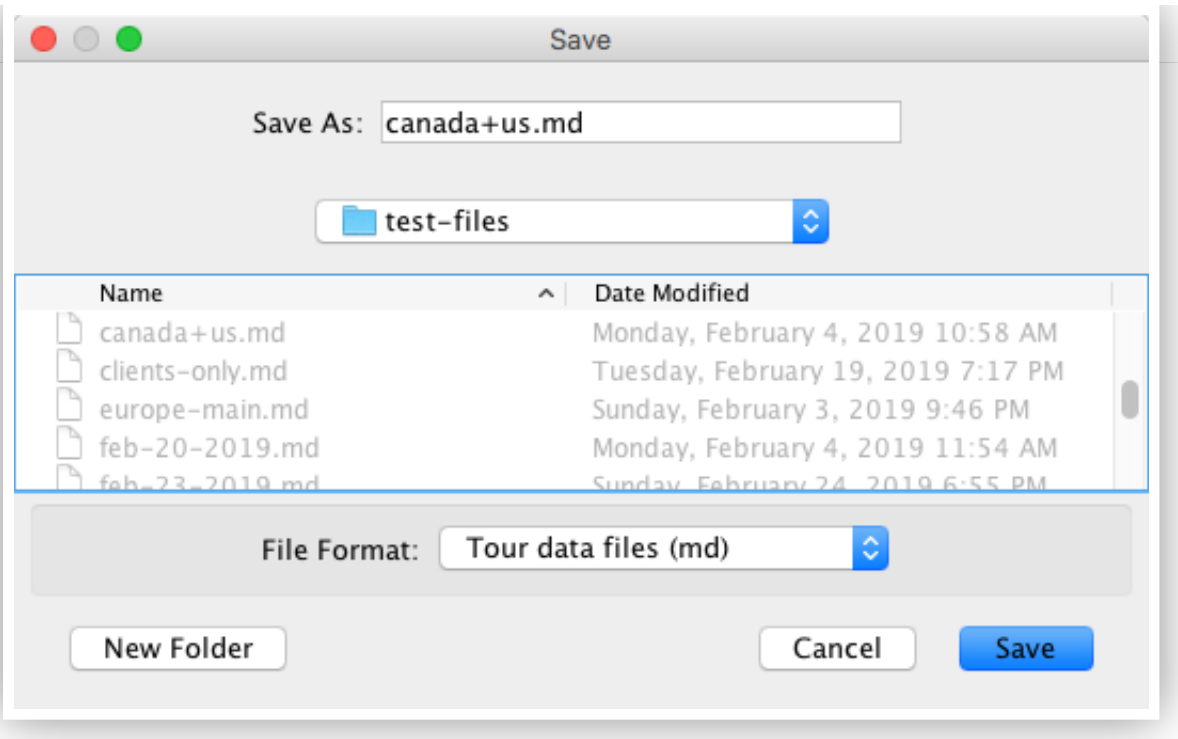
Use Case 3: Save data to a Wolf Travel data file

Preconditions: The program is running. The user has started a new Wolf Travel data file [UC4] or opened an existing one [UC2].

Main Flow: The user selects *File -> Save* to save the data to a file [S1][S2].

Subflows:

[S1] When the user selects *File -> Save,* a Save dialog opens to the path and filename used to open the file or used at the last file save operation [E1][E2]. The file system browser dialog opens with a filter on the default file extension (**.md*), as illustrated in Figure 6. The user enters the file name in the dialog and clicks **Save.** [E3][E4]



Save As: canada+us.md

test-files

Name	Date Modified
canada+us.md	Monday, February 4, 2019 10:58 AM
clients-only.md	Tuesday, February 19, 2019 7:17 PM
europe-main.md	Sunday, February 3, 2019 9:46 PM
feb-20-2019.md	Monday, February 4, 2019 11:54 AM
feb-23-2019.md	Sunday, February 24, 2019 6:55 PM

File Format: Tour data files (md)

Figure 6: File save dialog.

[S2] The data file is created, with each line of the file representing a client, a tour, or a reservation. The format for the lines is described in [UC2, S3]. The order of lines is as follows:

All clients are written first, ordered according to when they were read or added. Clients read from a file always come before clients added while the program was executing with the current dataset.

Tours and reservations are written next, ordered according to the description in [UC3, S4]. After each tour come the reservations for that tour, listed in order that the reservations were made. Reservations that were loaded from a file always come before reservations added while the program was executing with the current dataset.

The file sample.out.md is an example of a properly formatted file generated by the save process.

Alternative Flows:

[E1] If the user clicks **Cancel** on the file Save dialog, the dialog closes, aborting the file save operation and thus saving no data.

[E2] If the file has not been saved before, the File Save dialog opens to the default data file storage location (that is, ./) with the filename used to load or create new data file.

[E3] If the filename entered by the user is blank, just whitespace, or does not end with .md, an error dialog opens with the message "File not saved." The user clicks **OK,** closing the dialog and aborting the file save operation.

[E4] If an error occurs while writing the data file to disk, an error dialog opens. The user clicks **OK,** closing the dialog and aborting the file save operation.

Use Case 4: Open a new Wolf Travel data file

Preconditions: The program is running.

Main Flow: The user selects *File -> New* [S1]. The system saves the current data as needed [S2] and clears the text area displays of all data, leaving a blank slate for the user to enter new data [S3].

Subflows:

[S1] The user selects *File -> New* from the main menu [E1].
[S2] If the dataset currently in the system has been changed and not saved, a Save operation is completed [UC3], [E1].
[S3] The dataset currently in the system is cleared. All text areas on the GUI are cleared and all buttons are enabled.

Alternative Flows:

[E1] If the user opts not to save the data currently in the system, then the save operation is not completed.

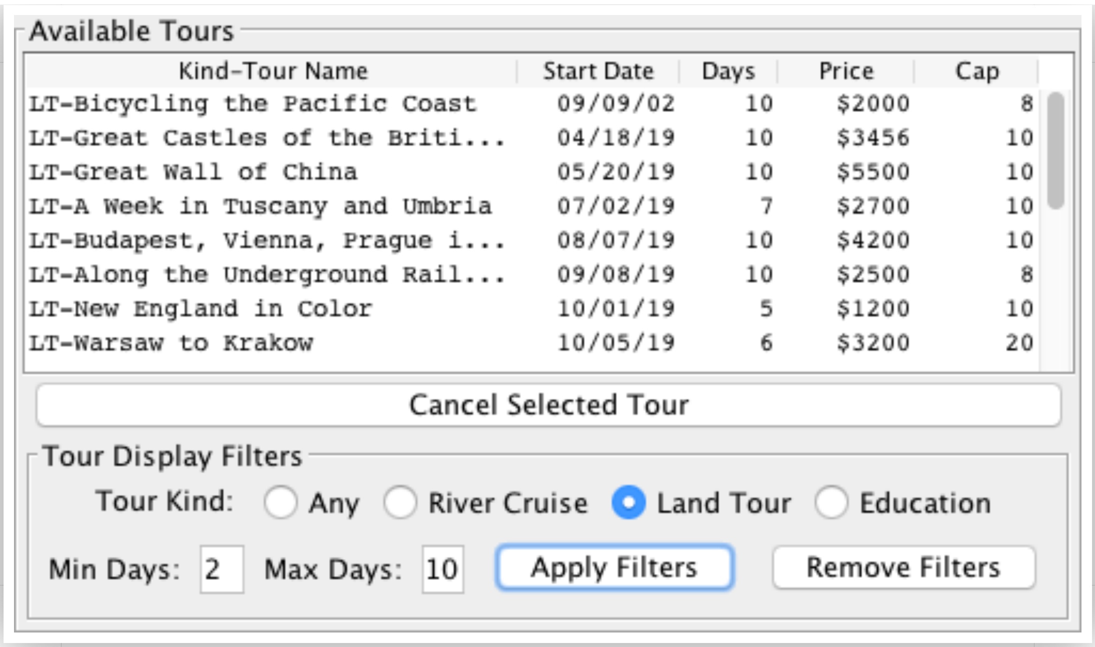
Use Case 5: Filter the tour display

Preconditions: The program is running and a file has been loaded [UC2] or a new file has been started [UC4].

Main Flow: The user filters the display of tours in the *Available Tours* panel [S1][S2], or the user removes the filters in order to display all tours [S3].

Subflows:

[S1] The user selects the radio button for the kind of tour to display, or selects *Any* not to filter by kind. In the *Min Days* and *Max Days* fields, the user optionally types the minimum duration and the maximum duration to restrict display to tours with durations falling into that range. If the minimum duration field is left blank, the minimum duration defaults to 0. If the maximum duration field is left blank, the maximum defaults to the maximum possible integer.
[S2] If the user clicks **Apply Filters,** then the display under *Available Trips* lists only the tours that meet all filters specified. No tour is shown as selected. [E1] Figure 7 below shows filtering the tour display to show only river cruises of 2 - 10 days duration.
[S3] If the user clicks **Remove Filters,** then the display under *Available Trips* lists all tours. No tour is shown as selected.


Available Tours

Kind-Tour Name	Start Date	Days	Price	Cap
LT-Bicycling the Pacific Coast	09/09/02	10	$2000	8
LT-Great Castles of the Briti...	04/18/19	10	$3456	10
LT-Great Wall of China	05/20/19	10	$5500	10
LT-A Week in Tuscany and Umbria	07/02/19	7	$2700	10
LT-Budapest, Vienna, Prague i...	08/07/19	10	$4200	10
LT-Along the Underground Rail...	09/08/19	10	$2500	8
LT-New England in Color	10/01/19	5	$1200	10
LT-Warsaw to Krakow	10/05/19	6	$3200	20

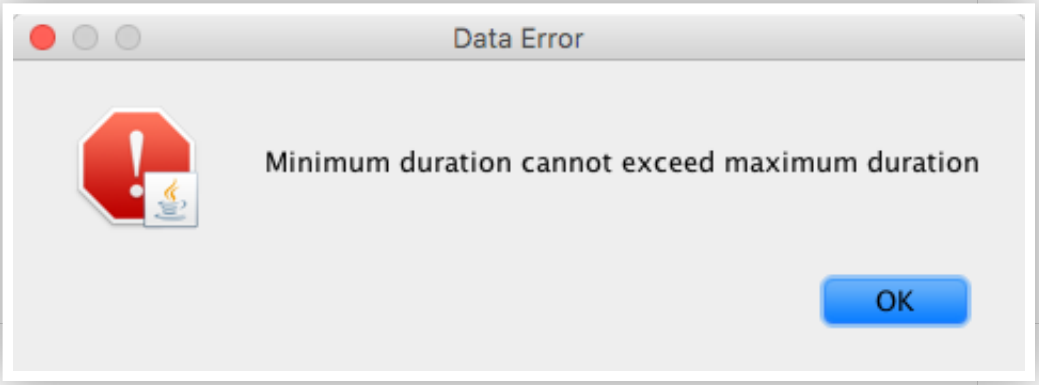
Tour Kind: Any, River Cruise, [✓ Land Tour], Education

Min Days: 2 Max Days: 10

Figure 7: Filtering tours according to River Cruise and durations of 2 - 10 days.

Alternative Flows:

[E1] If the user enters a minimum duration that is greater than the maximum duration, a dialog opens to inform the user and new filtering is not applied. See Figure 8 below.


Data Error

Minimum duration cannot exceed maximum duration

Figure 8: Error in setting minimum/maximum duration filters.

Use Case 6: Add a new client

Preconditions: The program is running [UC1] and a file has been loaded [UC2] or a new file has been started [UC4].

Main Flow: The user adds a new client using the *Add Client* widgets on the lower left of the GUI window [S1] [S2].

Subflows:

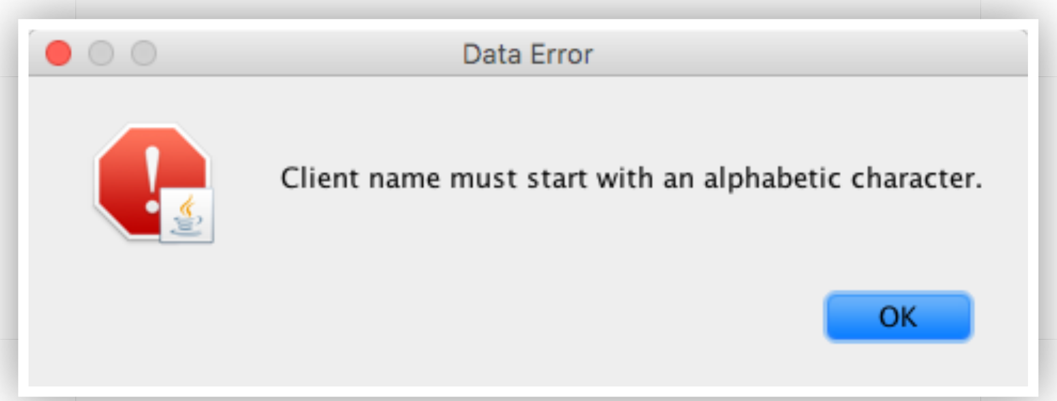
[S1] The user types the client's name in the *Client Name* field and types the client's contact information in the *Client Contact* field, then clicks **Add New Client**. [E1][E2][E3]

[S2] The system adds a new client with the given name and contact to the end of the *Clients* list. The new client is displayed in the same format as the others (name first, followed by contact information in parentheses). No client is selected, so no corresponding client reservations are displayed. The text fields for adding a client are cleared.

Alternative Flows:

[E1] If the user fails to type text for the client name or contact, a dialog opens informing the user that names and contacts cannot be blank. No client is added.

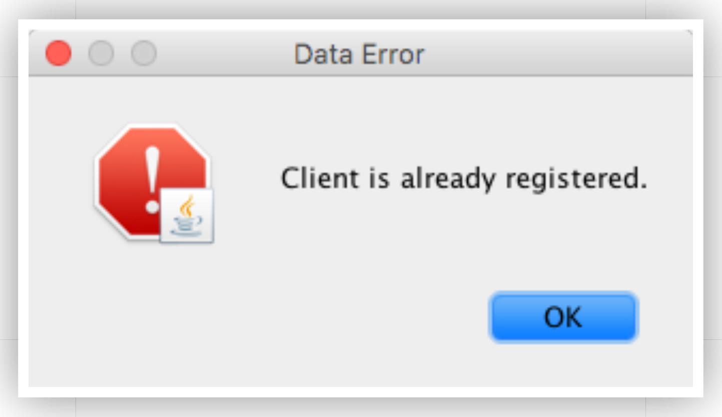
[E2] If the user types a name that does not begin with an alphabetic character or a contact that does not begin with an alphabetic or digit character, a dialog opens as shown in Figure 9. No client is added.



Client name must start with an alphabetic character.

Figure 9: Error attempting to add client whose first name starts with a non-alphabetic character.

[E3] If the user types name and contact information for an existing client (all comparisons are case insensitive), a dialog opens as shown below in Figure 10. No client is added.



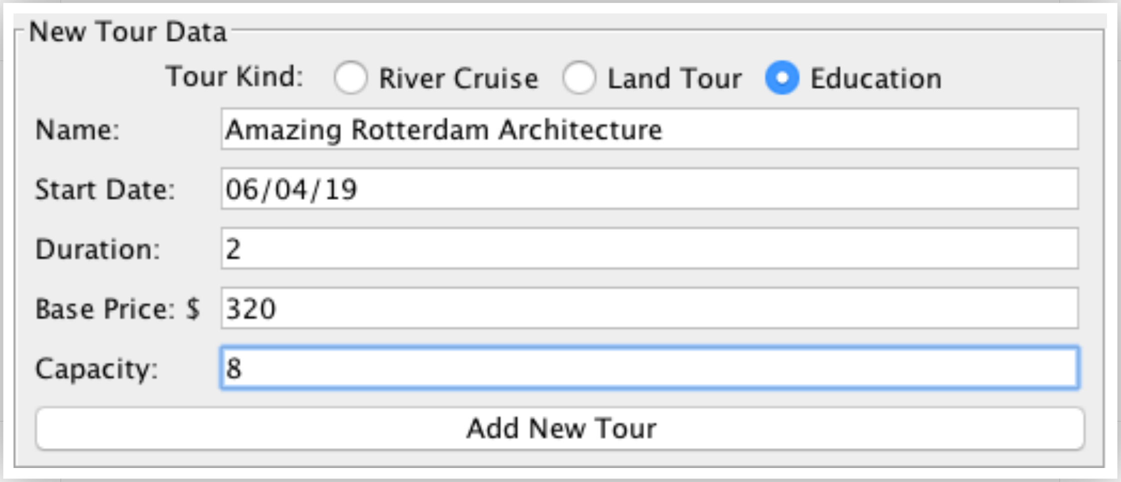
Client is already registered.

Figure 10: Error attempting to add a duplicate client.

Use Case 7: Add a new tour

Preconditions: The program is running [UC1] and a file has been loaded [UC2] or a new file has been started [UC4].

Main Flow: The user enters the new tour information in the widgets in the *New Tour Data* region of the GUI, then clicks **Add New Tour.** [S1-S7]
Figure 11 illustrates entry of new tour data.



New Tour Data

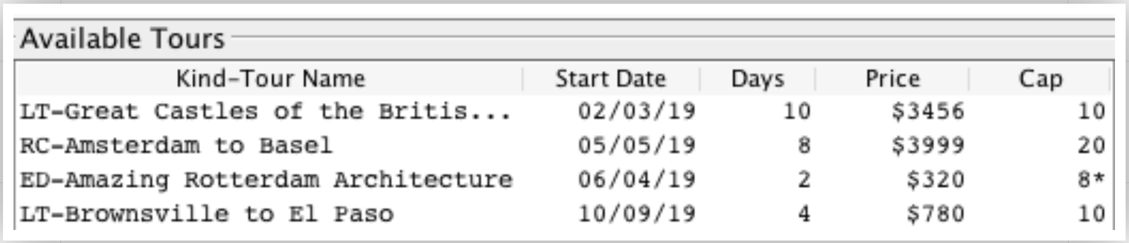
Tour Kind: ( ) River Cruise ( ) Land Tour [✓ Education]

Name:	Amazing Rotterdam Architecture
Start Date:	06/04/19
Duration:	2
Base Price: $	320
Capacity:	8

Figure 11: New tour input.

Subflows:

[S1] The user selects the tour kind.
[S2] The user types the tour name in the *Name* field.
[S3] The user types the tour start date in the *Start Date* field, using the format mm/dd/yy.
[S4] The user types the days duration (a positive integer) of the tour in the *Duration* field.
[S5] The user types the tour base price as an integer and with no leading '$' in the *Base Price* field.
[S6] The user types the tour capacity (a positive integer) in the *Capacity* field.
[S7] The user clicks **Add New Tour.** The tour is displayed in the proper position on the *Available Tours* list (according to current filters), where the ordering of the tours is as described in [UC2,S3]. Since every educational trip can double its capacity once, the capacity for any new educational trip is shown with a trailing asterisk, as illustrated in Figure 12. The input text fields are cleared and no tour is selected. [E1][E2]



Available Tours

Kind-Tour Name	Start Date	Days	Price	Cap
LT-Great Castles of the Britis...	02/03/19	10	$3456	10
RC-Amsterdam to Basel	05/05/19	8	$3999	20
ED-Amazing Rotterdam Architecture	06/04/19	2	$320	8*
LT-Brownsville to El Paso	10/09/19	4	$780	10

Figure 12: Available Tour list showing the newly added educational tour.

Alternative Flows:

[E1] If any entries described in [S2] through [S6] are not valid, a dialog opens to indicate the nature of the error and no new tour is added.
[E2] If the new tour matches an existing tour (same name ignoring case, same start date, and same duration), a dialog opens to indicate no duplicates are allowed and no new tour is added.

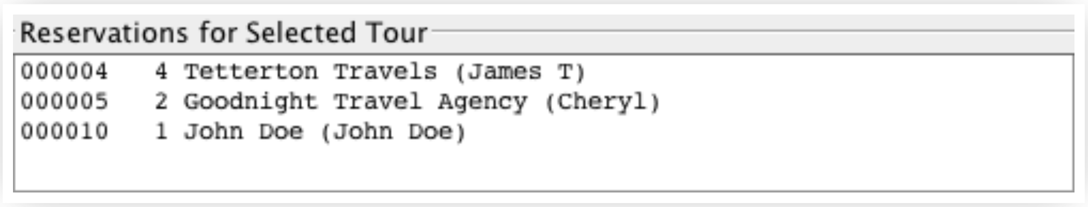
Use Case 8: View all reservations for a particular tour

Preconditions: The program is running [UC1] and a file has been loaded [UC2] or a new file has been started [UC4].

Main Flow: To select the tour of interest, the user clicks the tour in the *Available Tours* display. All reservations for the selected tour are listed under *Reservations for Selected Tour* [S1][S2].

Subflows:

[S1] The user optionally filters the *Available Tours* in order to reveal the desired tour [UC5], then clicks the tour.
[S2] The reservations are listed in the order that they were read or subsequently created. Each reservation listed is shown by confirmation code, then size of party, then client name and contact. The format should match the following example:
000004 4 Tetterton Travels (James T)
Figure 13 shows the reservations for the selected land tour from Figure 1 named "Along the Underground Railroad."


Reservations for Selected Tour

```
000004   4 Tetterton Travels (James T)
000005   2 Goodnight Travel Agency (Cheryl)
000010   1 John Doe (John Doe)
```

Figure 13: Reservations for a particular tour.

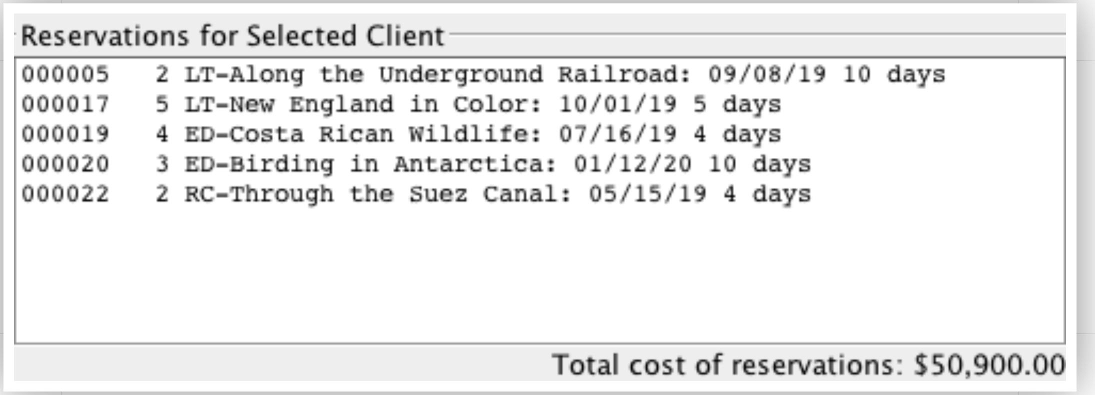
Use Case 9: View all reservations for a particular client

Preconditions: The program is running [UC1] and a file has been loaded [UC2] or a new file has been started [UC4].

Main Flow: The user clicks the client of interest in the *Clients* display. All reservations for the selected client are displayed under *Reservations for Selected Client* [S1][S2].

Subflows:

[S1] When the user clicks the client of interest, reservations for that client are displayed in the *Reservations for Selected Client* text area in the order that they were read or subsequently created. Each reservation is shown by confirmation code, then size of party, then tour information (kind, name, start date, duration). The format should match the following example:
000004 4 LT-Along the Underground Railroad: 09/08/19 10 days
[S2] The total cost of all the client's reservations is shown at the bottom of the *Reservations for Selected Client* area. Figure 14 shows the reservations for the selected client, with a total cost of $50,900.



```
000005   2 LT-Along the Underground Railroad: 09/08/19 10 days
000017   5 LT-New England in Color: 10/01/19 5 days
000019   4 ED-Costa Rican Wildlife: 07/16/19 4 days
000020   3 ED-Birding in Antarctica: 01/12/20 10 days
000022   2 RC-Through the Suez Canal: 05/15/19 4 days
```

Total cost of reservations: $50,900.00

Figure 14: Reservations for a particular client.

Use Case 10: Make a new reservation

Preconditions: The program is running [UC1] and a file has been loaded [UC2] or a new file has been started [UC4]. The current trip data includes at least one client and at least one tour.

Main Flow: The user selects the client and the tour for the reservation, types the number of people in the reservation party, and clicks **Make Reservation on Selected Tour for Selected Client** [S1]. The confirmation code is determined [S2], the cost is computed [S3], and the reservation is created and displayed under both *Reservations for Selected Tour* and *Reservations for Selected Client* [S4].

Subflows:

[S1] The user selects the client for the reservation from the *Client* list and the tour from the *Available Tours* list (filtering the *Available Tours* display as needed to select the tour [UC5]). The user types the number of people in the reservation party in the *Party size for new reservation* field then clicks **Make Reservation on Selected Tour for Selected Client** in the *Make/Change Reservations* panel of the GUI. [E1]
[S2] The confirmation code assigned to the reservation is 1 + the confirmation code of the largest confirmation code of a reservation in the current dataset. For example, if largest

code is 000025, then the code for the new reservation code would be 000026. Confirmation codes wrap at 999999. So if the largest code is 999999, then the new reservation would have code 000000.

[S3] The cost of the reservation depends on the kind of tour, the base price, and the party size.

For educational tours, the cost is base price X party size.

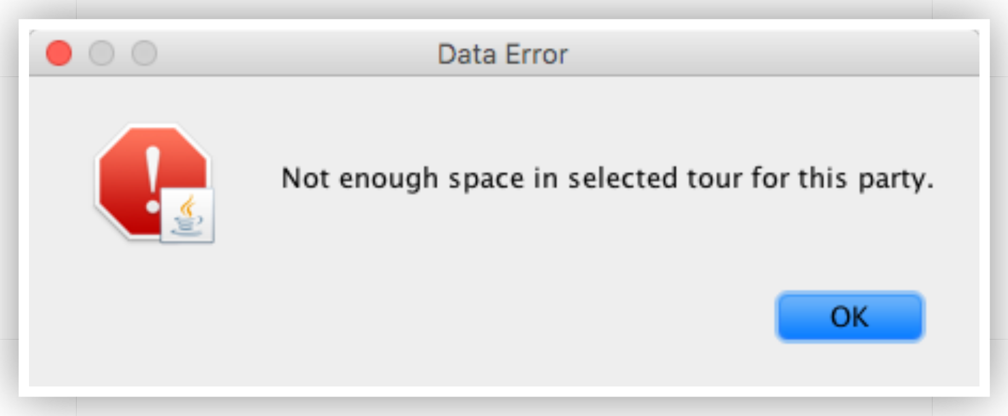
For land tours, the cost is base price X party size except for parties of size 1. For single-person parties, the cost is base price + 25% surcharge. For example, suppose a land tour has a base cost of $1000. Then the cost of a reservation for two people is $2000, but the cost of a party of one is $1250.

For river cruises, the cost is base cost X number of people in party for parties of an even number of people. For parties of an odd number of people, the cost is the same except for a 50% surcharge for one of the people. For example, suppose a river cruise has a base cost of $1000. Then the cost of a reservation for 4 people is $4000, but the cost of a party of five people is $5500.

[S4] The new reservation is displayed at the bottom of the *Reservations for Selected Tour* list (illustrated in Figure 13) and at the bottom of the *Reservations for Selected Client* list (illustrated in Figure 14). The cost of reservations for the selected client is updated to include the cost of the newly created reservation.

Alternative Flows:

[E1] If adding the new reservation would result in reservations for more people than the tour can accommodate, then a dialog opens indicating a capacity exception, as illustrated in Figure 15. The reservation is not made.


Data Error

Not enough space in selected tour for this party.

Figure 15: Attempt to book over capacity.

Use Case 11: Cancel a reservation

Preconditions: The program is running [UC1] and a file has been loaded [UC2] or a new file has been started [UC4]. The current dataset for the program includes at least one reservation.

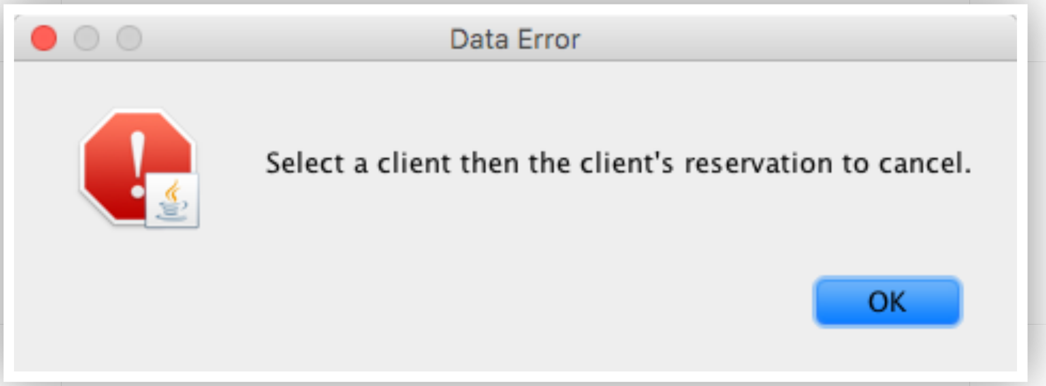
Main Flow: The user selects the client for the reservation then selects the reservation from the client's corresponding list of reservations and clicks **Cancel Selected Reservation** [S1]. The client's reservation list is updated, as is the list of reservations for the corresponding tour [S2].

Subflows:

[S1] The user selects the relevant client from the *Client* list and the reservation to be canceled from the *Reservations for Selected Client* list. Then the user clicks **Cancel Selected Reservation,** which is a button on the *Make/Change Reservations* panel. [E1]
[S2] The reservation is removed from the *Reservations for Selected Client* display, and the total cost of the client's reservations is decreased by the cost of the canceled reservation. When the tour for the canceled reservation is selected, then the canceled tour no longer appears on the *Reservation for Selected Tour* list.

Alternative Flows:

[E1] If the user does not select a client then a reservation (as shown on the *Reservations for Selected Client* list, an error dialog opens to indicate that no reservation was selected, as illustrated in Figure 16.



Select a client then the client's reservation to cancel.

Figure 16: Attempt cancel a reservation without selecting which reservation to cancel.

Use Case 12: Cancel a tour

Preconditions: The program is running. The current dataset for the program includes at least one tour.

Main Flow: The user selects the tour to be canceled then clicks **Cancel Selected Tour.** [S1]. The selected tour is removed from the system along with all reservations for that tour [S2].

Subflows:

[S2] The user optionally filters the tour display so that the tour to be canceled is listed among the *Available Tours* [UC5], then clicks **Cancel Selected Tour,** which is the button immediately below the *Available Tours* display.

[S2] The tour is removed from the system, along with all reservations for that tour (clients no longer have reservations for the canceled tour). The GUI then shows no selected client and no selected tour.

Alternative Flows:

[E1] If the user does not select a tour to be canceled, an error dialog opens to indicate that no tour was selected.

Use Case 13: Shutdown

Preconditions: The program is running.

Main Flow: The user closes the main window or selects *File -> Exit* [S1]. The program checks to be sure the current data has been saved [S2], then quits execution [S3].

Subflows:

[S1] The user closes the window to exit the program or selects *File -> Exit* to exit the program [E1].
[S2] If the current data file has changed since it was opened or last saved, the user is prompted to save the file [UC3].
[S3] The program stops execution with no errors or exceptions.

Design

Trying to jump from requirements right into coding without doing some design first is bound to lead to trouble. You will eventually implement our teaching staff design (Project 2 Part 2), but first you need to propose your own. To do this, we give you a scenario and insist on some restrictions for your design.

Scenario

With these requirements, you can now propose a design of an application that will help your boss submit the winning bid to Wolf Travel! Your design must describe the static layout of the system by identifying the objects, their state and behavior, and the relationships among the objects needed to implement the requirements.

Assignment

You must design an application that satisfies the requirements of the Wolf Travel application. Your design must be described in document containing a design rationale and a UML class diagram.

Your design must:

- utilize the Model-View-Controller (MVC) design pattern (see the note about MVC, below)
- utilize one additional design pattern
- contain at least one interface or abstract class
- contain at least one inheritance relationship
- contain at least one composition relationship
- contain one *custom* ArrayList
- contain one *custom* LinkedList

To help your boss evaluate your design, you should answer the following technical questions in your design document as part of the rationale:

1. What objects are important to the system's implementation and how do you know they are important?
2. What data are required to implement the system and how do you know these data are needed?
3. Are the responsibilities assigned to an object appropriate for that object's abstraction and why?
4. What are the relationships among objects (such as inheritance and composition) and why are those relationships are important?
5. Have you identified any design patterns appropriate for implementing the system (such as the State Pattern)? How are the identified patterns appropriate for implementing the system?
6. What are the limitations of your design? What are the constraints of your system?

MVC Note

Java Swing, the user interface (UI) libraries for Java, does not follow the strictly traditional definition of MVC. Instead, Java Swing utilizes what might be called a *separable model architecture*. This means that the model is separate and distinct from the view/controller classes that make up the UI. For your design, you will focus on the Model. In your UML class diagram, you should represent the UI as a class with no state or behavior. Your diagram should also show which class(es) your UI will interact with through some type of composition/aggregation/association relationship. The relationship between your model and the UI ***must*** be justified in your design rationale.

When thinking about the relationships between your UI and the model, consider the following questions:

1. What are the data and behaviors of your model that will be shown through the UI?
2. How does your UI get those data to display; what methods of the model must be called?

Format

Submit your design proposal via Gradescope under the assignment named **P2P1: Design**.

Use the provided design proposal template as a starting point for your design proposal. Use a UML diagramming tool (options are listed the Software Development Practices notes) to create your UML diagram. Incorporate your UML diagram into your written proposal (using an editing tool such as MS Word) and save the entire document as a PDF. Alternatively, create a PDF for the design proposal and another PDF for the UML diagram, then append the diagram to the proposal to make a single PDF document.

Need a little more direction?

See the following example design proposal: Sample Design Proposal.

Testing

This project requires you to do white box and black box testing. You can defer white box testing until Part 2 of this project. But now, you need to prepare some black box test cases. We will start you off with a scenario.

Scenario

Your manager has requested a black box test plan that describes five test cases that will determine if the finished program is ready to accompany the bid. Your manager wants you to write the tests before you begin development to clarify the inputs to and outputs from the system. Each test must demonstrate that the program satisfies a scenario from the requirements. A scenario is one major path of functionality as described by the requirements.

Assignment

You will write at least five (5) tests for the Wolf Travel project. Use the provided black box test plan template to describe your tests. Each test must be repeatable and specific; all input and expected results values must be concrete. All inputs required to complete the test must be specified. Additionally, you must provide instructions for how a tester would set up, start, and run the application for testing (What class from your design contains the main method that starts your program? What are the command line arguments, if any? What do the input files contain?) Describe the instructions at a level where anyone using Eclipse could run your tests.

Sample test input file

Format

You must submit your black box test plan via your section's submission platform as a PDF. Use the provided template as a starting point for your black box test plan. Save your BBTP as a PDF. Submit the PDF to Gradescope under the assignment named **P2P1: BBTP**.

Need a little more direction?

See the following example black box test plan: Sample Black Box Test Plan.

Deployment

For this class, deployment means submitting your work for grading. For Part 1 of this programming assignment, you must submit two PDF documents:

1. Design document with incorporated UML class diagram.
2. Black box test plan document.

Make sure that your submissions satisfy the grading rubrics.

You should submit the documents for Project 2 Part 1 to Gradescope.

Submission Reminders

The electronic submission deadline is precise. Do not be late. You should count on last minute system failures (your failures, ISP failures, or Gradescope failures). You are able to make multiple submissions of the same file. (Later submissions to a Gradescope assignment overwrite the earlier ones.) To be on the safe side, start submitting your work as soon as you have completed a substantial portion.

Project 2, Part 1: Wolf Travel Tours ▶