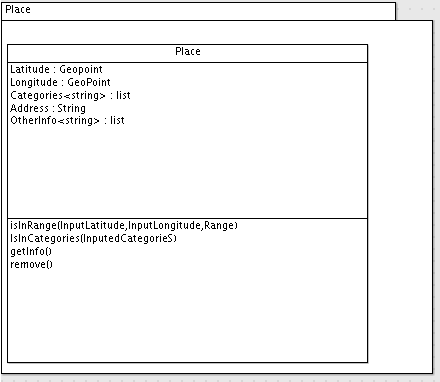
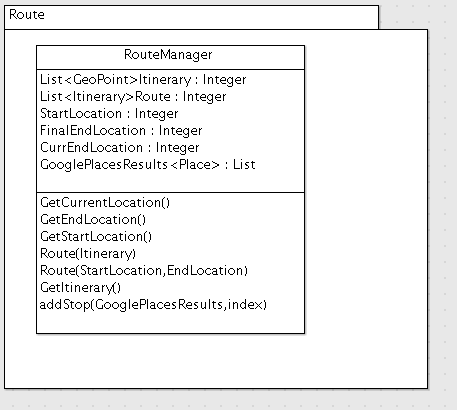
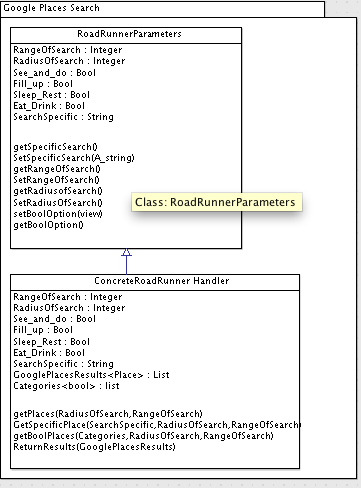
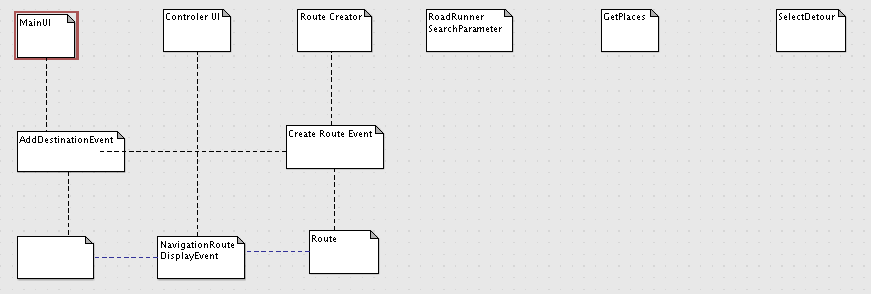
Static Class Diagrams

We presented the different packages separately in order to fit the diagram on the width of a page.

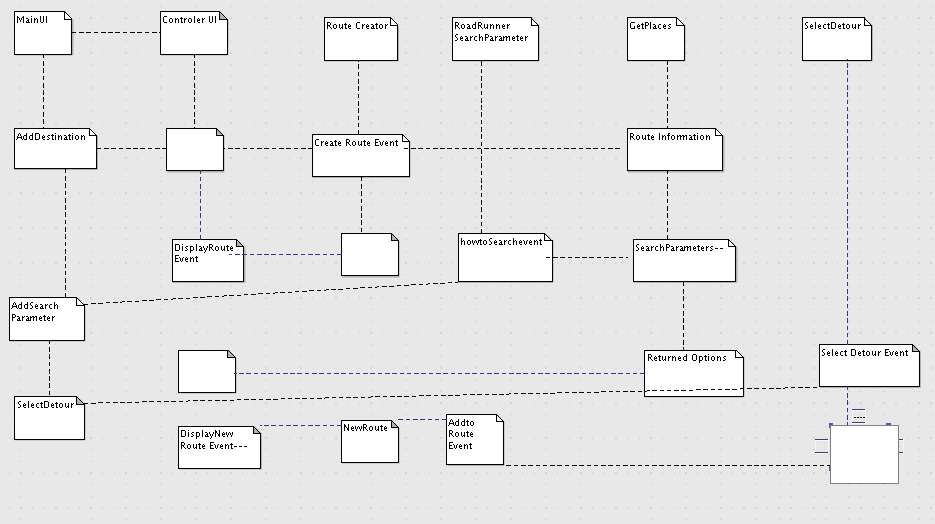




Sequence diagrams

The first Sequence Diagram is for the user to just use the RoadRunner as a simple navigation. They Open up RoadRunner, they enter their destination, the route is created and navigation information is returned.

The second event is that of when a user will add in search parameters to their route. They add their destination and a route is returned and displayed. Then they add the search parameters which will pass its search parameters along with the route to the generate options event. This returns the options to the user where they select an option and this option is passed to the create route which returns a new route and displays it



**Design Approach**

We included most of the main design concepts except for polymorphism. We used Object class concept since we since deal with the information in the form of classes, such as the Places information. We used information hiding design pattern because the user never really interacts with the information after they input their search parameters, they just interact with results. Our main design is based upon the implementation of the MapQuest map functions. We used this because it has a lot of the navigation information already created and easy to understand. We have also tried to create the code in a way that each item is being implemented separately in a Module style of way. We did this because when we are testing we can easily remove certain elements in order to test the rest of the service. ­We also use inheritance in our navigation functionality since each map is an extension of a lower level of the map class. In other words the route display inherits a lot of the information that the class uses based on the simple map. The design pattern that we most used was the Chain of Reasonability design pattern. This is something we used because the client dealt with a handler and this handler based upon the information into the handler (the roadrunner search window) concrete handlers (the Google places algorithm and search style) is called up and generates results that are returned to the handler. The handler itself never executes what is going to happen based on its data.

Submit your design approach (\*\*must be object-oriented) and any design patterns you plan on using in your code. Approach should be clearly articulated and mention at least one design pattern.

Contribution Summary

Daniel Campos-Created the Sequence Diagrams and worked upon the creation of the UI and the route management and creation.

Zoe Konrad-Did contribution summary, design approach and Tested and created a lot of the UI.

Daniel Yoon-Worked upon research and documentation for our project and created some of the navigation aspects of the application.

Nick Marton- Create Most of the UI and worked upon the Location services and the places services.

Status Report

Achieved

Basic Road Runner App Design and Diagrams

Basic Map Application and Demos

Nonfunctioning Road Runner App Empty shell

Merged Map and Road Runner App

Basic Good UI

Basic Settings Adjustment

Basic design to display information of the Places but not working.

Outputting locations of places

Trip Status Partially working but without some of the real-time parts

Partial Merging of the information of places and the app

Debugged app that is accurate

Risks:

Had to change some of the deliverables as mentioned in the appendix

By using the MapQuest API vs the Google one we will have to change the way that we display some information and because we don’t have access to a lot of data we will have to change what we will display.

We are not working with as effective as information as we originally wished

Plan:

Make the app less buggy by working on it and using our bug tracker to see our changes

Test our app a lot with friends and family to make sure it is easy to use and understandable

Functionality to divide and search all the locations on route based on specific parameter or categories(Specification #3)

Bug Testing on backend to make sure that the category and search and functioning properly and fully

Preset distance markers(Specification #3)

Debug so that all the information is visible and correct

Appendix

After we had done a large amount of development we figured out that we would not be able to deliver the navigation we wanted by Google maps due to changes in their policy of the information they display. We were forced to transition to use a separate format, in this case using MapQuest. This caused up to not be able to do Real Time navigation. In other words MapQuest does not allow the route to actively update while you are driving, much like the original Google maps applications. We also since we will not have up to date information we will remove the trip status and information window because it does not provide any useful information. If we find a work around or a fix then we will include it.