The class diagram breaks the components into 4 overall packages based on functionality and category: trip planning, billing, serialization, and people. The classes themselves follow a convention: all attributes are either private or read-only, methods (including constructors) are limited to the minimum required accessibility.   
 There were several design choices that were driven by the UI, such as the PaymentType and PersonType enums. These are perhaps unnecessary when the objects they refer to are solely utilized internally, but they present a great method of collecting the different types into a usable structure, for say a printed list of options in a CLI or a drop-down in a GUI. Though this UML modelling tool restricts the possibility, the enums will most likely carry a reference to the class it is associated with so they can be instantiated using the enums. For example, PersonType.Traveller.getNewInstance() might be used instead of a strict factory. There are also some enums that could be used even more like ordinary classes, such as the TransportType enum. This could be used to store the different values associated with each transport type, e.g. speed, for easy access throughout the program.