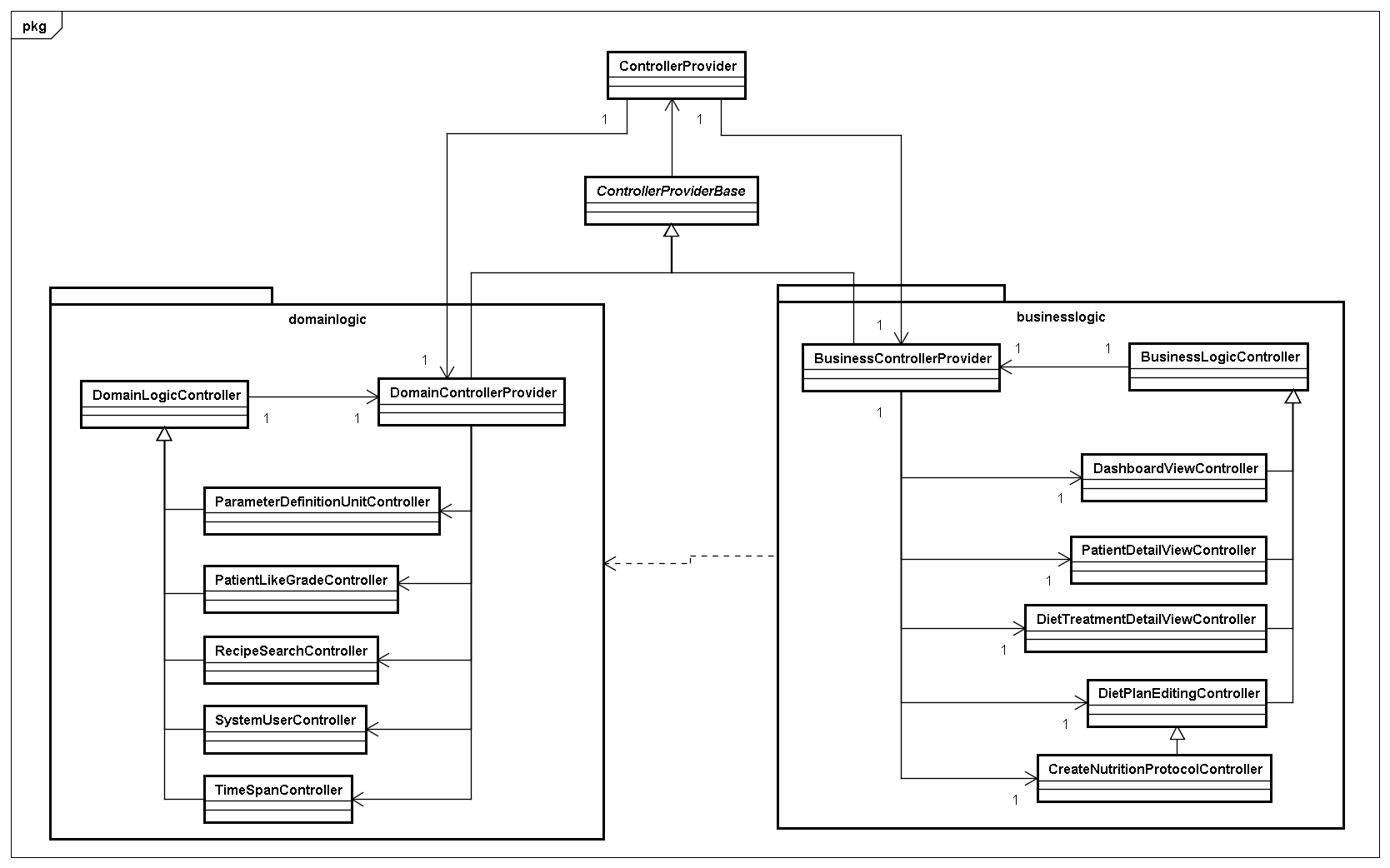
Session Basierte Controller

Eines der Hauptprobleme bei der Entwicklung eines Web-basierten Usecases war das Anbieten von Controllern welche pro aktiver Browsersitzung ein einziges Mal verfügbar sind. JSF bietet zwar die Möglichkeit an, Klassen als ManagedBeans eine Instanz pro Session zu halten, doch wäre die Markierung von Controllern als Session-Beans View-Logik in der Business-Logik Schicht, was in der bestehenden Architektur nicht erlaubt ist.

Um ein Session-Singleton Verhalten zu erzeugen und die Kommunikation zwischen den Controllern zu ermöglichen, wurde folgende Klassenstruktur realisiert: 

Eine Klasse welche von *ControllerProviderBase* erbt. Bietet eine Liste von Controllern an. In EasyDiet gibt es zwei solche Implementationen:

* Der *BusinessLogicProvider* bietet Singletons für alle verfügbaren Anwendungslogik-Controller an.
* Der *DomainLogicProvider* bietet Singletons für alle verfügbaren Domänenlogik-Controller an.

Alle Controller einer Schicht (Anwendungslogik, Domänenlogik) erben von einer gemeinsamen Basisklasse für Controller welche die Navigation auf den zugehörigen Provider ermöglicht. Dadurch wird der Zugriff auf die Controllerinstanzen innerhalb derselben Schicht ermöglicht.

Die Klasse *ControllerProvider* hält Instanzen der verfügbaren *ControllerProviderBase* Implementationen. Somit sind in einer Instanz der Klasse *ControllerProvider* Singletons aller verfügbaren Controller verfügbar. Da die *ControllerProviderBase* Klasse den Zugriff auf den übergeordneten *ControllerProvider* ermöglicht, können die Controller über die Schichten hinweg miteinander kommunizieren.

Um nun innerhalb der Web-Anwendung pro Session alle Controller zur Verfügung zu haben, hält nun ein JSF session-scoped Managed-Bean eine Instanz eines solchen *ControllerProvider.* Dadurch kann in der Präsentationsschicht über das sog. *ControllerBean* auf alle in der Session gültigen Controller zugegriffen werden.