Die Klassen innerhalb des Projekts wurden in drei verschiedene Pakete eingeteilt. Auch die Klassendiagramme wurden zur besseren Übersichtlichkeit in drei Diagramme aufgeteilt. Trotzdem bestehen natürlich Beziehungen zwischen den Klassen der unterschiedlichen Pakete, die zum einen durch die Namensgebung als auch durch die untenstehenden Beschreibungen deutlich werden.

1. standardPackage

*Klassendiagramm\_Fachklassen*

Das erste Paket beinhaltet alle Fachklassen, die im Rahmen des Webshops benötigt wurden. Im Verlauf des Projekts haben sich gegenüber dem Entwurf im Fachkonzept einige Änderungen ergeben. Die neue Klasse „FestivalSuchobjekt“ wurde erstellt, da im Bereich der Suche (Ticket Shop) zu einem Festival zusätzliche Informationen, wie beispielsweise ein Maximalpreis gespeichert werden muss. Um die Übergabe an die JSP möglichst einheitlich zu gestalten, wird dann ein FestivalSuchobjekt, das alle nötigen Informationen, die dem Anwender in der Ticket-Suche angezeigt werden müssen, beinhaltet, übergeben.   
Darüber hinaus wurde das Kann-Kriterium mit der Einbindung von festivalübergreifenden Artikeln realisiert. Da zu diesen auch Bilder verfügbar sein sollten, wurde in der Klasse „Artikel“ das Attribut „bildpfad“ hinzugefügt. Außerdem wurde für die Realisierung des Kann-Kriteriums mit der dreimaligen Sperre bei falscher Passworteingabe das zusätzliche Attribut „passwortZaehler“ bei der Klasse „Benutzer“ hinzugefügt.

2. managerPackage

*Klassendiagramm\_Managerklassen*

Die Manager-Klassen beinhalten alle Funktionen, um direkt auf die Datenbank zugreifen zu können. Die Klasse „Datenbankverbindung“ ist dabei dafür zuständig, die Kommunikation mit der Datenbank zu steuern. Das bedeutet, dass in dieser Klasse sowohl die Erstellung der Verbindung als auch jeweils eine vorgefertigte Methode für das Durchführen von Select-, Insert- und Update-Befehlen existiert. Diese Schnittstellen werden in den tatsächlichen Manager-Klassen verwendet und bieten den Vorteil, dass dort nur der SQL-Befehl erstellt werden muss und die zurückgelieferten Ergebnisse (bei einem Select-Aufruf beispielsweise ein Result-Set) verarbeitet werden müssen. Der technische Teil zum Verbindungsaufbau ist damit in einer Klasse gekapselt.   
Damit die Anwendung bei einer Nichtverfügbarkeit der Datenbank trotzdem benutzerfreundlich reagiert, wurde die „DatenbankException“ erstellt. Diese liefert dem Anwender in einem solchen Fall eine angemessene Fehlerseite und protokolliert den Fehler im Log.

Die Managerklassen nutzen die Fachklassen aus dem ersten Paket. Bei der Suche nach einem Artikel wird das zurückgegebene Ergebnis (Result-Set) verarbeitet, indem eine Instanz der entsprechenden Fachklasse erstellt wird, in der die Informationen dann abgelegt werden. Wenn mehrere Ergebnisse enthalten sind, wird einsprechend eine ArrayList vom Typ der Fachklasse erstellt.

Es wurden sprechende Namen verwendet, sodass beispielsweise die Klasse „ArtikelManager“ alle Methoden zur Erstellung, Ändern und Löschen von Artikeln beinhaltet. Das bedeutet, dass hier vorrangig die Fachklasse Artikel instanziiert wird. In einigen Fällen, wie beispielsweise bei dem „BestellungsManager“ werden auch weitere Klassen als nur die Fachklasse „Bestellung“ instanziiert. Da das Objekt Bestellung unter Anderem aus einer Liste mit Bestellpositionen besteht, wird hier auch die Klasse „Bestellposition“ instanziiert.

Zur Übersichtlichkeit folgt der Aufbau aller Manager-Klassen innerhalb des Pakets dem erläuterten Prinzip.

3. servletPackage

*Klassendiagramm\_Servletklassen*

Im Rahmen des MVC-Konzepts ist die Verwendung von Servlets zur Annahme und Beantwortung von Anfragen des Clients erforderlich. Im Klassendiagramm wurde auf die Darstellung der post()- und get()-Methoden verzichtet. Jedes Servlet enthält diese Methoden zur Übermittlung der benötigten Daten. Die Servlets arbeiten sowohl mit den Fachklassen (erstes Paket) als auch mit den Managerklassen (zweites Paket). Die Ermittlung eines Benutzers über die ID ist damit beispielsweise nur ein Aufruf einer Methode innerhalb des „BenutzerManagers“ und die Speicherung des Rückgabewerts in einer Instanz der Klasse „Benutzer“.   
Um nicht für jede Anfrage ein neues Servlet verwenden zu müssen, wurde der Parameter „aktion“ verwendet. Dadurch werden in der Regel einzelne Use-Cases gemeinsam in einem Servlet bearbeitet. Das Servlet „Artikelverwaltung“ beinhaltet zum Beispiel die Möglichkeiten zur Anlage, Änderung, etc. von Artikeldaten durch den Administrator. Zur Anlage muss der Wert des Parameters in diesem Fall „anlegen“ haben. Dadurch entsteht eine logische Kapselung von Funktionalitäten, die vor Allem der Übersichtlichkeit dient. Der Parameter „aktion“ wird innerhalb des gesamten Projekts verwendet. Lediglich die Servlets Login, Logout und Registrierung werden nicht über diesen Parameter gesteuert. Dies liegt daran, dass die Servlets bereits sehr viele Überprüfungen beinhaltet und bei weiteren Funktionalitäten zu unübersichtlich werden würden.   
Generell besitzen durch den oben genannten Aufbau alle Servlet-Klassen eine einheitliche Struktur.