**1.State pattern**

State Pattern je dinamička verzija Strategy paterna. Objekat mijenja način ponašanja na osnovu trenutnog stanja. Ovaj Pattern se postiže promjenom podklase unutar hijerarhije klasa.

U našem projektu moguće je iskoristiti ovaj patern za praćenje putnika i njihovu interakciju sa Etf Railwaysom. Tako za putnike koji su putovali određeni broj puta imamo specijalne ponude i popuste, a za putnike koji prvi put sarađuju sa Etf Railwaysom nema specijalnih ponuda i popusta, te oni plaćaju punu cijenu.

**2.Observer pattern**

Uloga Observer paterna je da uspostavi relaciju između objekata tako kada jedan objekat promijeni stanje, drugi zainteresirani objekti se obavještavaju.

U našem projektu, ovaj patern je iskoristen prilikom kupovine karte, gdje korisnik ima mogućnost da izabere opciju da mu se šalju obavijesti putem eMaila o svim novim ponudama i vožnjama Etf- Railwaysa.

**3.Interpreter pattern**

Interpreter pattern služi za evaluiranje gramatike nekog jezika i jezičkih izraza. Implementira se AbstractExpression interface koji koristi kompozitni pattern da bi riješio neki jezički problem.

Obično se koriste TeminalExpression i CompoundExpression klase koje služe za rekurzivno rješavanje problema interpretacije nekog jezičkog izraza gdje je CompoundExpression neki dio pravila koji poziva drugo pravilo za obradu izraza, dok je TerminalExpression bazni slučaj za neko pravilo.

U našem projektu bi ovaj patern mogli iskoristiti za provjeravanje validnosti passworda prilikom registracije, ili validnost username(dovoljan broj znakova, korišteni dozvoljeni znakovi...).