

Domain model using Java SE

```

EntityManagerFactory emf = Persistence.createEntityManagerFactory("sadh2dm");
EntityManager em = emf.createEntityManager();
EntityTransaction tx = em.getTransaction();
tx.begin();
////////////////////////////////////////
Product p = Product.newSpreadsheet("Test");
em.persist(p);
Product result = em.find(Product.class, p.getProductid());
assertEquals("Product id must be equals",p.getProductid(),
result.getProductid());
////////////////////////////////////////
tx.commit();
em.close();

```

Domain model with service layer using EJB

```

long p_id = pManager.newProductSpreadsheet("SpreadSheet by Cop");
long c_id = cManager.newContract(p_id, 1000, "2015-01-20");
cManager.calRecognitions(c_id);
Contract searchResult = cManager.findContract(c_id);
assertEquals("Date from save = from find", (new MfDate("2015-01-20")).toSqlDate(),
searchResult.getWhenSigned().toSqlDate());
assertEquals("Money must equals", Money.dollars(1000).amount(),
searchResult.recognizedRevenue(new MfDate("2015-04-24"))
.amount());
assertEquals("Money rev = all rev recog", searchResult.getRevenue().amount(),
searchResult.recognizedRevenue(new MfDate("2015-04-24")).amount());

```

Domain model with service layer using Spring

```

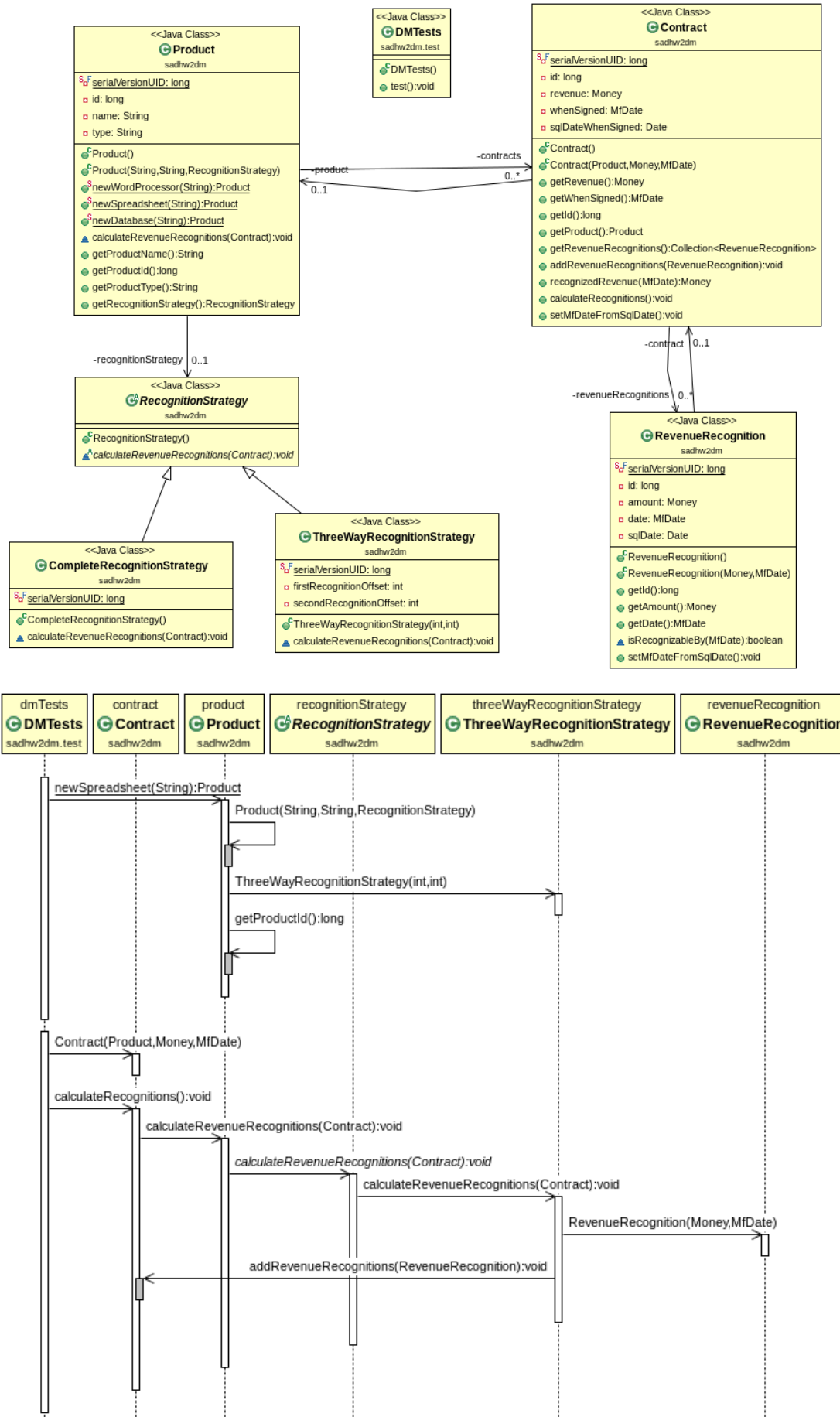
long p4 = pService.newProductSpreadsheet("SpreadSheet 2");
long c1 = cService.newContract(p4,1000,"2015-01-20");
cService.calRecognitions(c1);
Contract searchResult = cService.findContract(c1);
assertEquals("Date from save = from find", (new MfDate("2015-01-20")).toSqlDate(),
searchResult.getWhenSigned().toSqlDate());
assertEquals("Money must equals", Money.dollars(1000).amount(),
searchResult.recognizedRevenue(new MfDate("2015-04-24"))
.amount());
assertEquals("Money rev = all rev recog", searchResult.getRevenue().amount(),
searchResult.recognizedRevenue(new MfDate("2015-04-24")).amount());

```

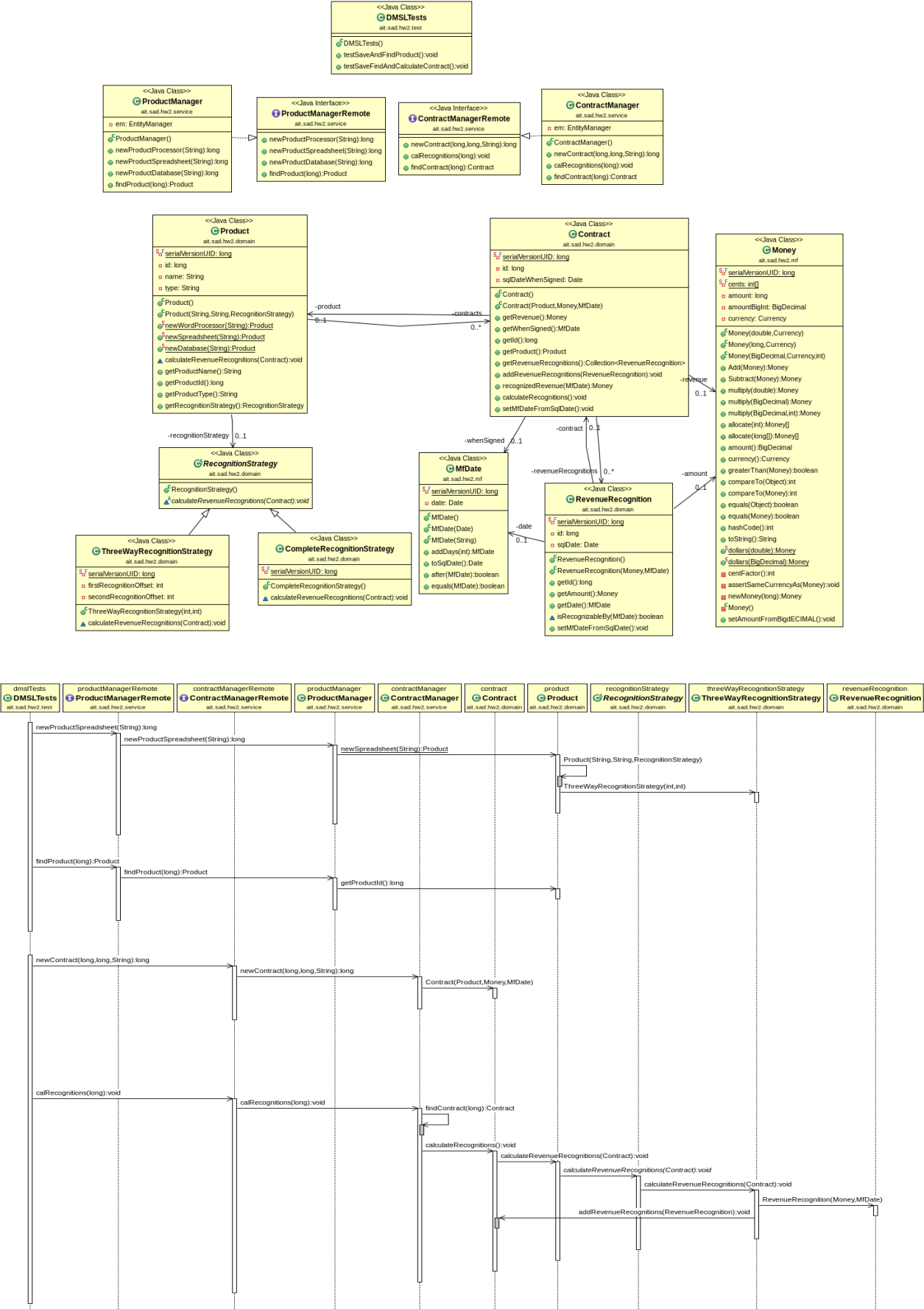
Summary

The difference between this homework and last homework is hw1 use ArrayList to collect data, but this hw2 use the real Database. In this homework2, the difference between the first method and other 2 is the service layer. The one with service layer control all the transactions in the service layer and do not let the users know what really happen inside. If the system got crashed, the system will handle all conflicts automatically and be sure that all data is corrected. On the other way, the one without service layer have to handle the transaction themselves. This can make system conflict easily. The difference between last 2 method is the setting up the environment part. For the idea of working is the same. So that's why, their sequence diagram look the same. By using the Spring instead of EJB, Spring will be more easy to setup the environment. Thats make chances to got mistake will be increased.

Domain model using Java SE



Domain model with service layer using EJB



Domain model with service layer using Spring

