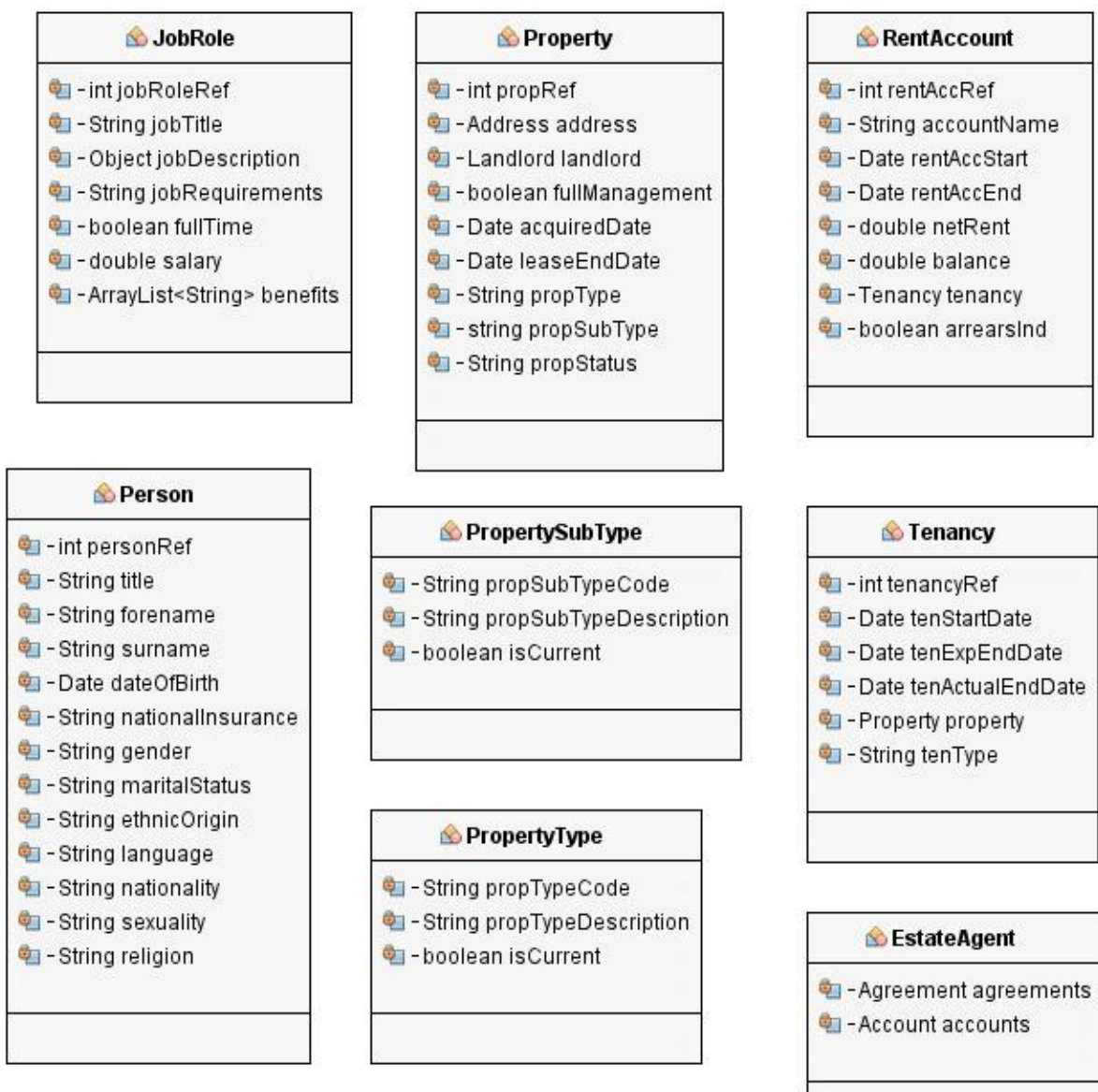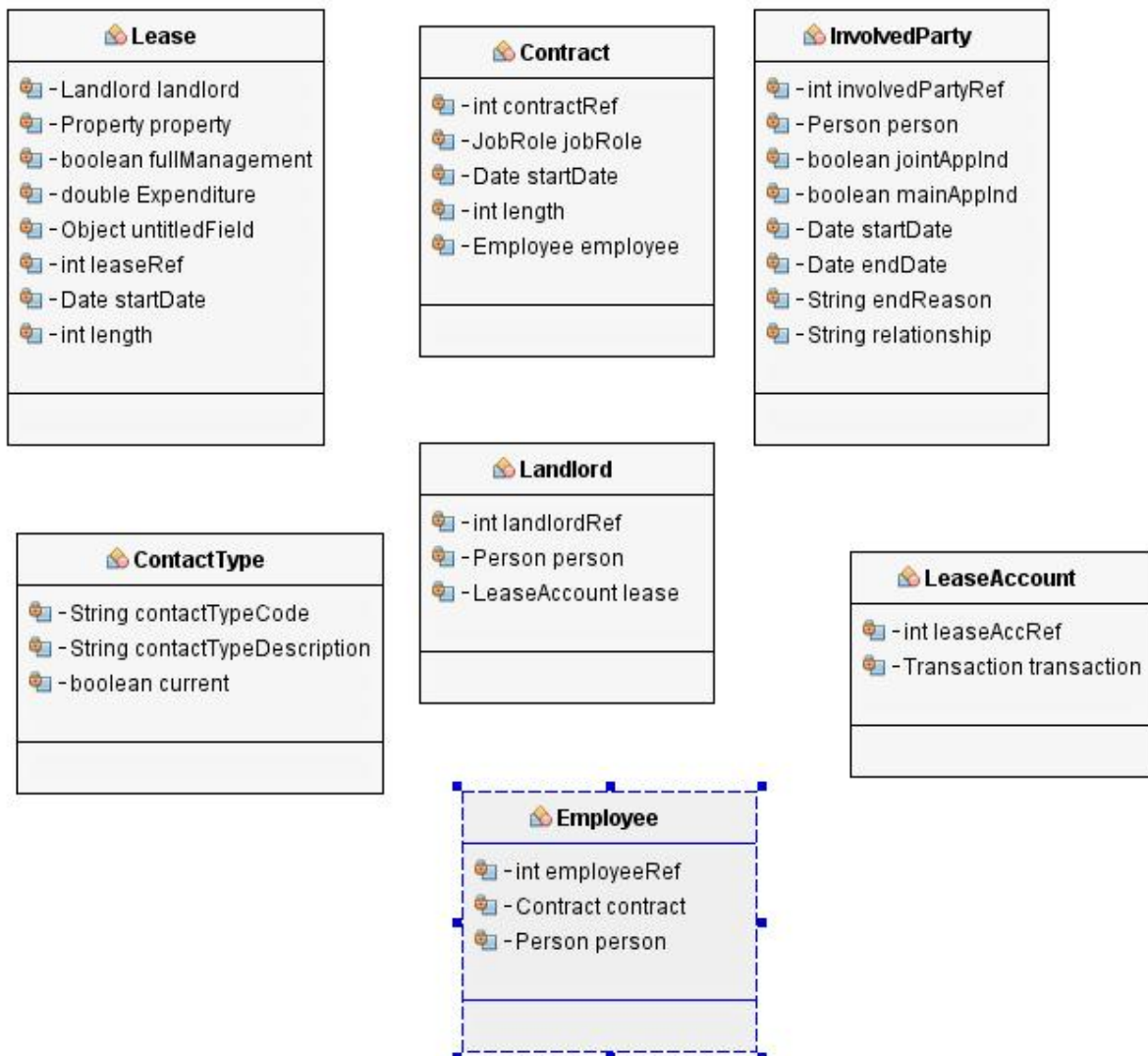# Appendices B

Class Diagram

Due to adopting the agile software lifecycle method, the program evolved during the development as I moved through each iteration, however I have now updated the class diagram due to the agile method principles. Also due to the size of the system, and lack of rescources it was difficult for me to get the whole class diagram along with relationships in one sheet so I have broken it down into pieces and is below:

JobRole

- int jobRoleRef
- String jobTitle
- Object jobDescription
- String jobRequirements
- boolean fullTime
- double salary
- ArrayList<String> benefits

Property

- int propRef
- Address address
- Landlord landlord
- boolean fullManagement
- Date acquiredDate
- Date leaseEndDate
- String propType
- string propSubType
- String propStatus

RentAccount

- int rentAccRef
- String accountName
- Date rentAccStart
- Date rentAccEnd
- double netRent
- double balance
- Tenancy tenancy
- boolean arrearsInd

Person

- int personRef
- String title
- String forename
- String surname
- Date dateOfBirth
- String nationalInsurance
- String gender
- String maritalStatus
- String ethnicOrigin
- String language
- String nationality
- String sexuality
- String religion

PropertySubType

- String propSubTypeCode
- String propSubTypeDescription
- boolean isCurrent

PropertyType

- String propTypeCode
- String propTypeDescription
- boolean isCurrent

Tenancy

- int tenancyRef
- Date tenStartDate
- Date tenExpEndDate
- Date tenActualEndDate
- Property property
- String tenType

EstateAgent

- Agreement agreements
- Account accounts

Lease

- Landlord landlord
- Property property
- boolean fullManagement
- double Expenditure
- Object untitledField
- int leaseRef
- Date startDate
- int length

Contract

- int contractRef
- JobRole jobRole
- Date startDate
- int length
- Employee employee

InvolvedParty

- int involvedPartyRef
- Person person
- boolean jointApplnd
- boolean mainApplnd
- Date startDate
- Date endDate
- String endReason
- String relationship

Landlord

- int landlordRef
- Person person
- LeaseAccount lease

ContactType

- String contactTypeCode
- String contactTypeDescription
- boolean current

LeaseAccount

- int leaseAccRef
- Transaction transaction

Employee

- int employeeRef
- Contract contract
- Person person

Class Descriptions

Server classes:

- AccountImpl – This class represents an account for 'MSc Properties' and implements Account from the common package.

- AddressImpl – This class represents an address for a property and implements Address from the common package.

- AddressUsageImpl – This class represents an actual usage of an address by a person and implements AddressUsage from the common package.

- AgreementImpl – This class represents an agreement for 'MSc Properties' between an office and a client and implements Agreement from the common package.

- ApplicationImpl – This class represents an application to 'MSc Properties' for private rented accommodation and implements Application from the common package.

- ContactImpl – This class represents a contact for a person or an office and implements Contact from the common package.

- ContractImpl – This class represents a contract between 'MSc Properties' and an employee, and extends AgreementImpl from the server package and implements Contract from the common package.

- Database – This class represents the application database that will hold all of the system data and although this cycle did not include the object relational mapping functionality, during a later cycle it will manage the connection to the MySQL database and deal with queries to the MySQL database to create, update and delete data within the database.

- ElementImpl – This class represents a system element, such as a religion or title for a person and implements Element from the common package.

- EmployeeImpl – This class represents an employee of 'MSc Properties and implements Employee from the common package.

- EmployeeAccountImpl – This class represents an account for an employee contract set up by an 'MSc Properties' office and extends AccountImpl from the server package and implements EmployeeAccount from the common package.

- InvolvedPartyImpl – This class represents a household member of an application for private rented accommodation and implements InvolvedParty from the common package.

- JobRoleImpl – This class represents a job role for an employee of 'MSc Properties' and implements JobRole from the common package.

- JobRoleBenefitImpl – This class represents a benefit for the associated job role of 'MSc Properties' and implements JobRoleBenefit from the common package.

- LandlordImpl – This class represents a landlord of a property 'MSc Properties' manage, and implements Landlord from the common package.

- LeaseImpl – This class represents a lease between 'MSc Properties' and a landlord, and extends AgreementImpl from the server package and implements Lease from the common package.

- LeaseAccountImpl – This class represents an account for a landlord lease set up by an 'MSc Properties' office and extends AccountImpl from the server package and implements LeaseAccount from the common package.

- ModifiedByImpl – This class represents a modification to a system object, such as an update to a property, and implements ModifiedBy from the common package.

- NoteImpl – This class represents a note for a system object, and implements Note from the common package.

- OfficeImpl – This class represents an office of 'MSc Properties' and holds the Agreements and Accounts associated with the office, and implements Office from the common package.

- PersonImpl – This class represents a person within the 'MSc Properties' system, and can be associated with an employee, landlord or involved party and implements Person from the common package.

- PropertyImpl – This class represents a property that 'MSc Properties' did or does manage and implements Property from the common package.

- PropertyElementImpl – This class represents an element of a property, for example rent or number of bedrooms and implements PropertyElement from the common package.

- RentAccountImpl - This class represents an account for an application tenancy set up by an 'MSc Properties' office and extends AccountImpl from the server package and implements RentAccount from the common package.

- TenancyImpl - This class represents a tenancy between 'MSc Properties' and an involved party of an application, and extends AgreementImpl from the server package and implements Lease from the common package.

- TransactionImpl – This class represents a transaction for an account of 'MSc Properties', and implements Transaction from the common package.

- ServerImpl – This class represents the controller class of the model, and although this cycle did not include the networking functionality, during a later cycle it will act as the actual server, and deal with setting up the remote server for clients to connect to.