INVENTORY MANAGEMENT SYSTEM

By Patryk Maryn

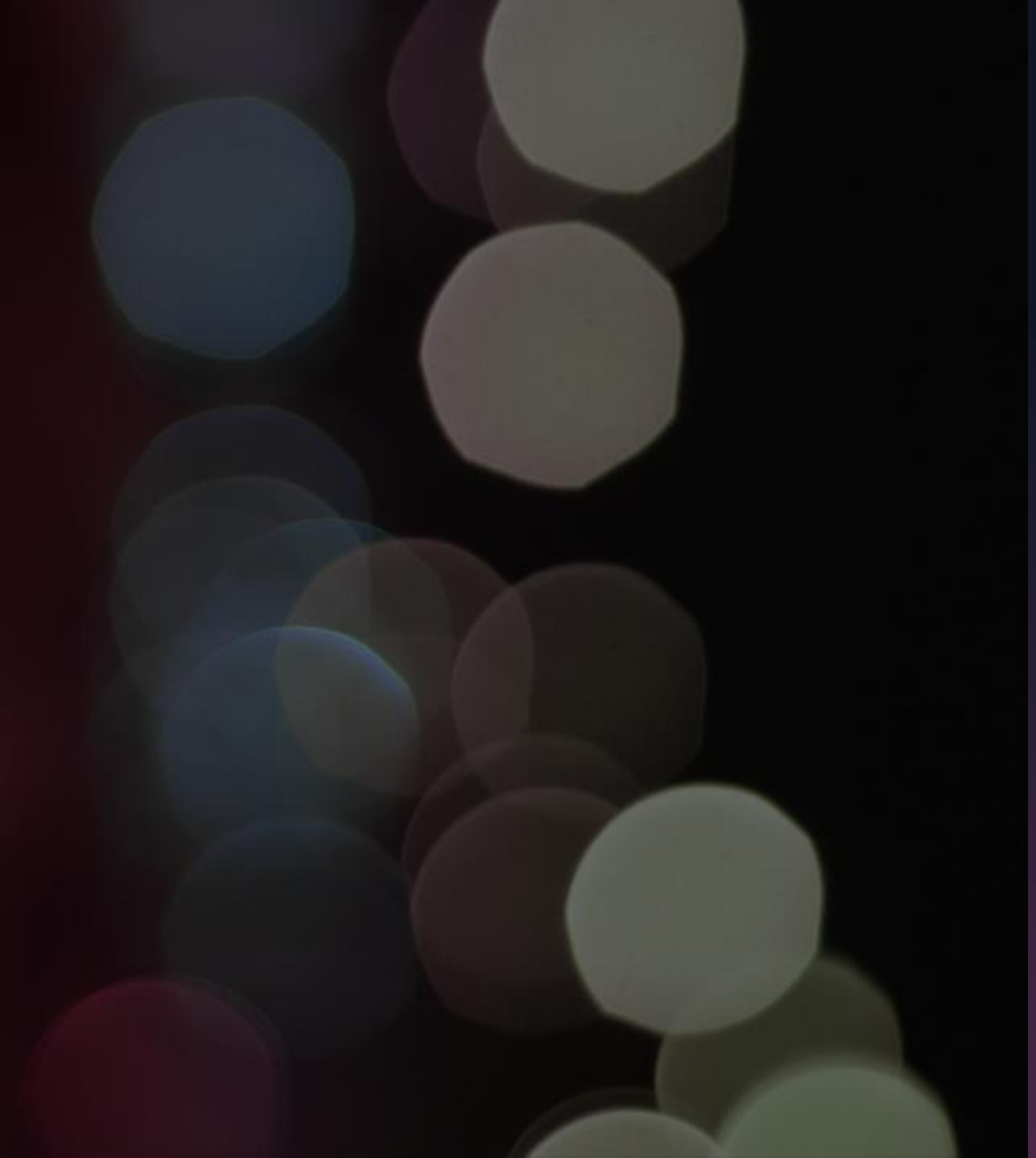
SPECIFICATION

CREATE A PROJECT WITH UTITISATION OF TOOLS AND METHODOLOGIES:

- AGILE & PROJECT MANAGEMENT (KANBAN BOARD)
- RELATIONAL DATABASE (MYSQL 5.7)
- CLOUD FUNDAMENTALS (GOOGLE GCP)
- REPOSITORY MANAGEMENT (NEXUS REPOSITORY)
- ~~VERSION CONTROL SYSTEM (GIT)~~
- AUTOMATED TESTING (JUNIT AND MOCKITO)

APPLICATION THAT USES JAVA AS PROGRAMMING LANGUAGE TO INTERACT WITH DATABASE SENDING

CREATE, READE, UPDATE, DELETE REQUESTS BY USER COMMAND LINE INPUTS AND DISPLAYING RESULTS IN COMMAND LINE



SCOPE:

- Kanban board
- Relational database
- Functional application
- Unit tests and integration tests
- Version control repository

WHAT I HAVE LEARNT

Lesson 1

Understanding the project requirements is the key, Not doing so may cause bad design decisions

Lesson 2

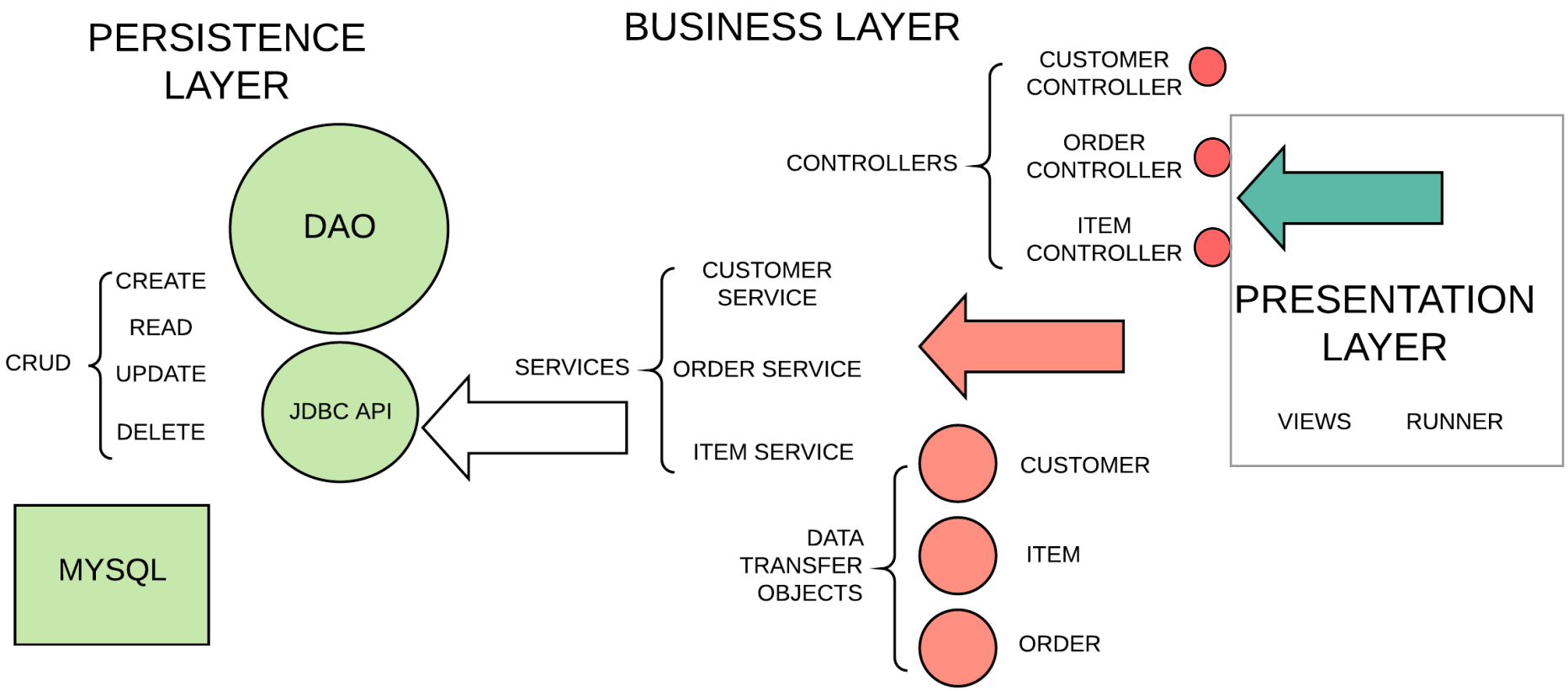
It's crucial to deliver demo or sample project to the client as soon as possible to find out if the project is on the good track

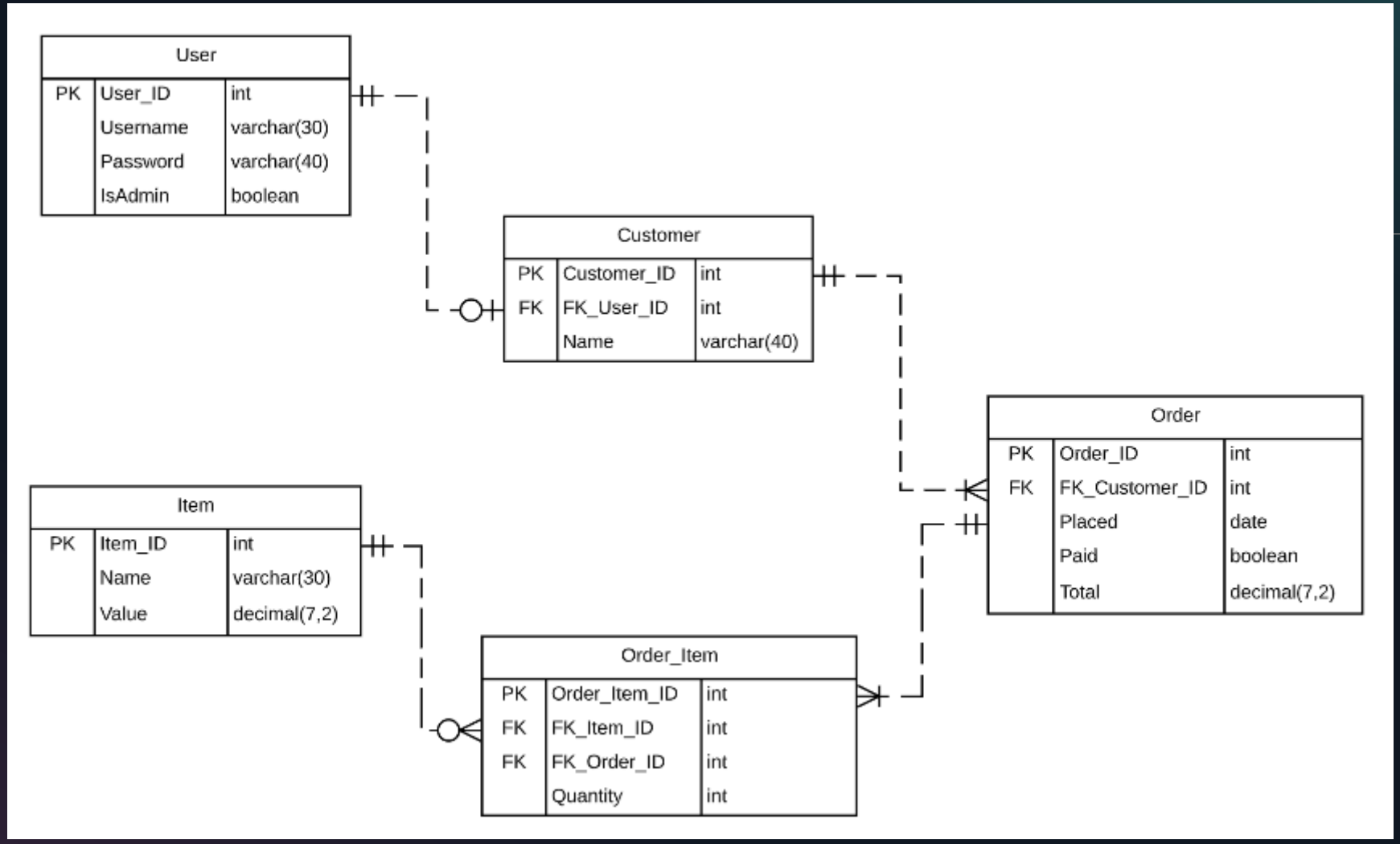
Lesson 3

When designing classes following SOLID principles. Reliance on interfaces rather than class object will reduce dependency and make code easier to test and maintain

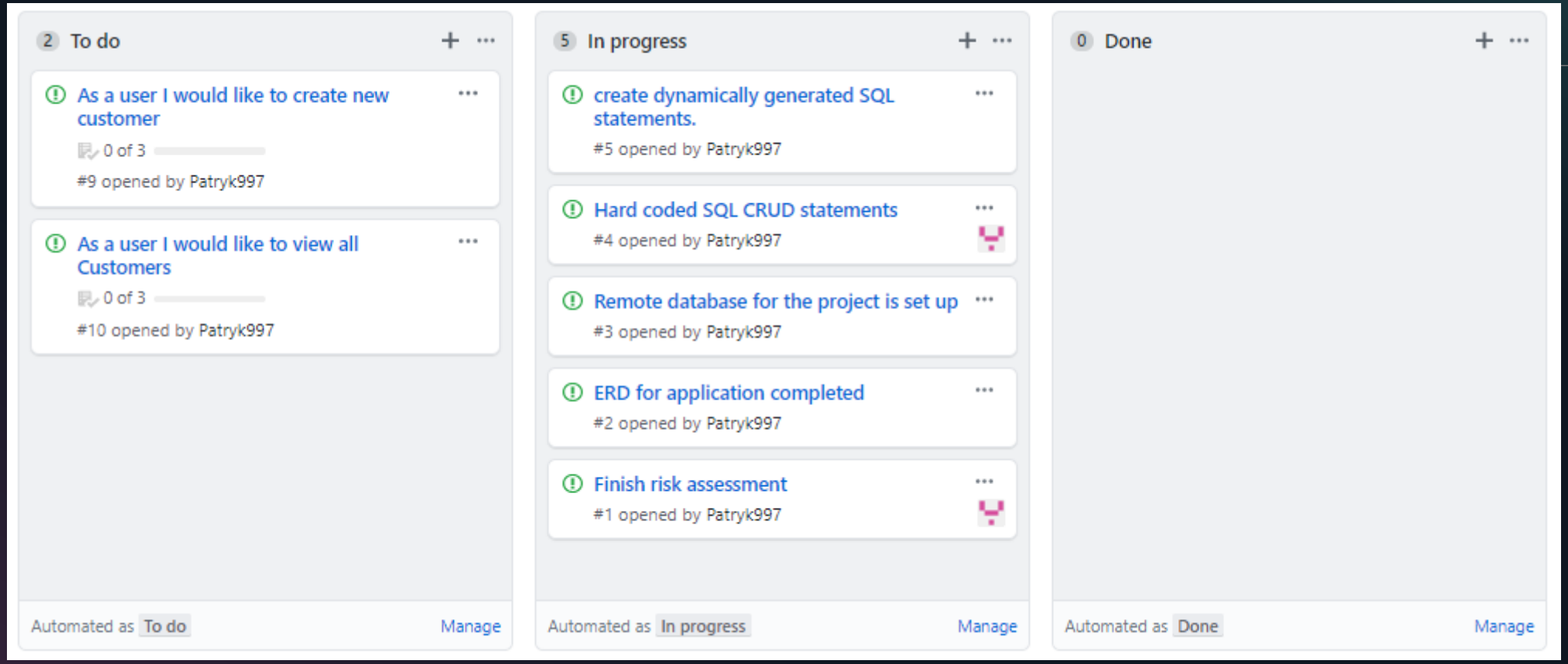
Lesson 4

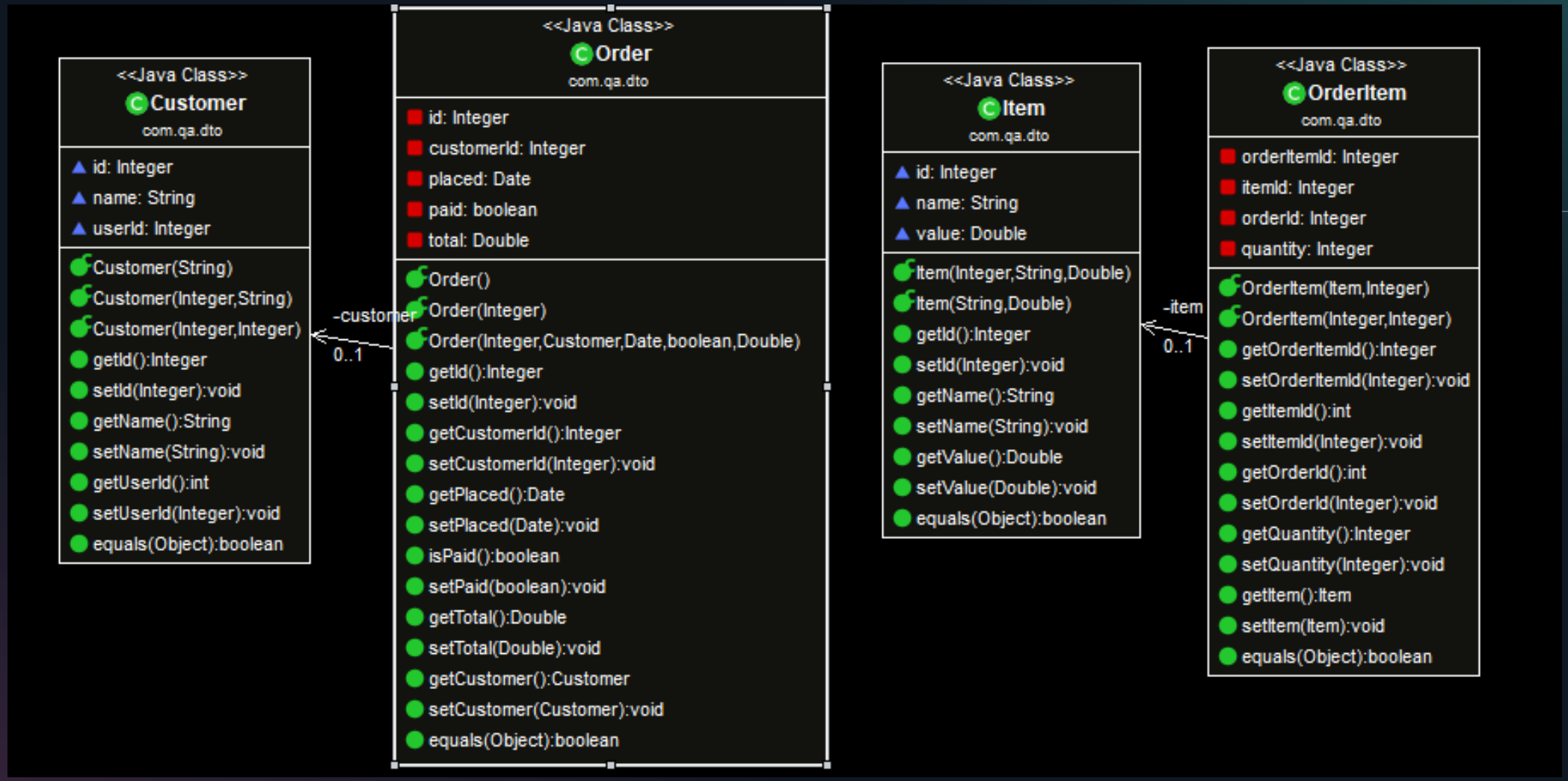
Uniqueness of software may cause difficulties to estimate the schedule. However most of the problems have been solved already.

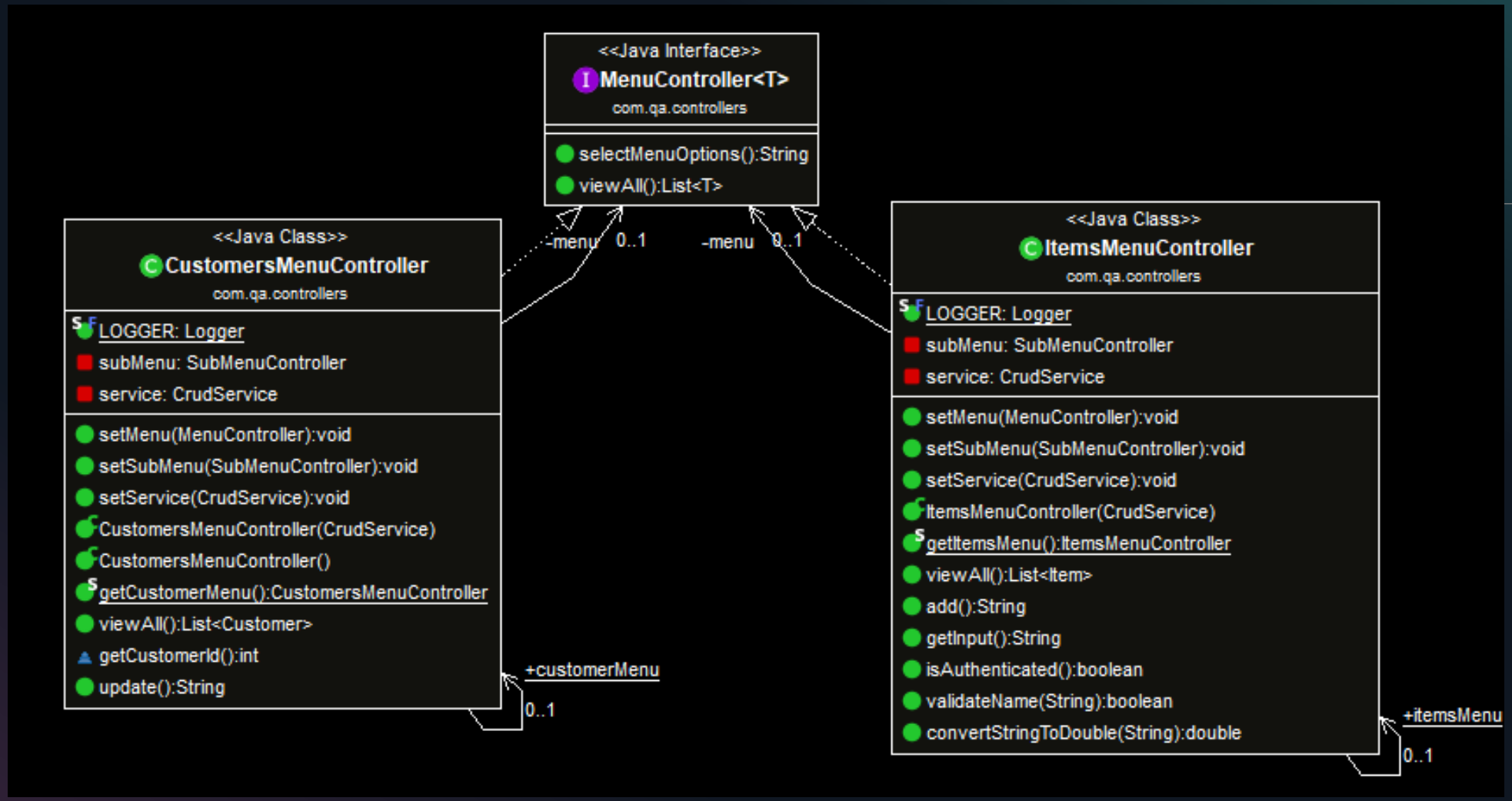

PERSISTENCE LAYER
DAO
CRUD
CREATE
READ
UPDATE
DELETE
JDBC API
MYSQL
BUSINESS LAYER
SERVICES
CUSTOMER SERVICE
ORDER SERVICE
ITEM SERVICE
CONTROLLERS
CUSTOMER CONTROLLER
ORDER CONTROLLER
ITEM CONTROLLER
DATA TRANSFER OBJECTS
CUSTOMER
ITEM
ORDER
PRESENTATION LAYER
VIEWS
RUNNER

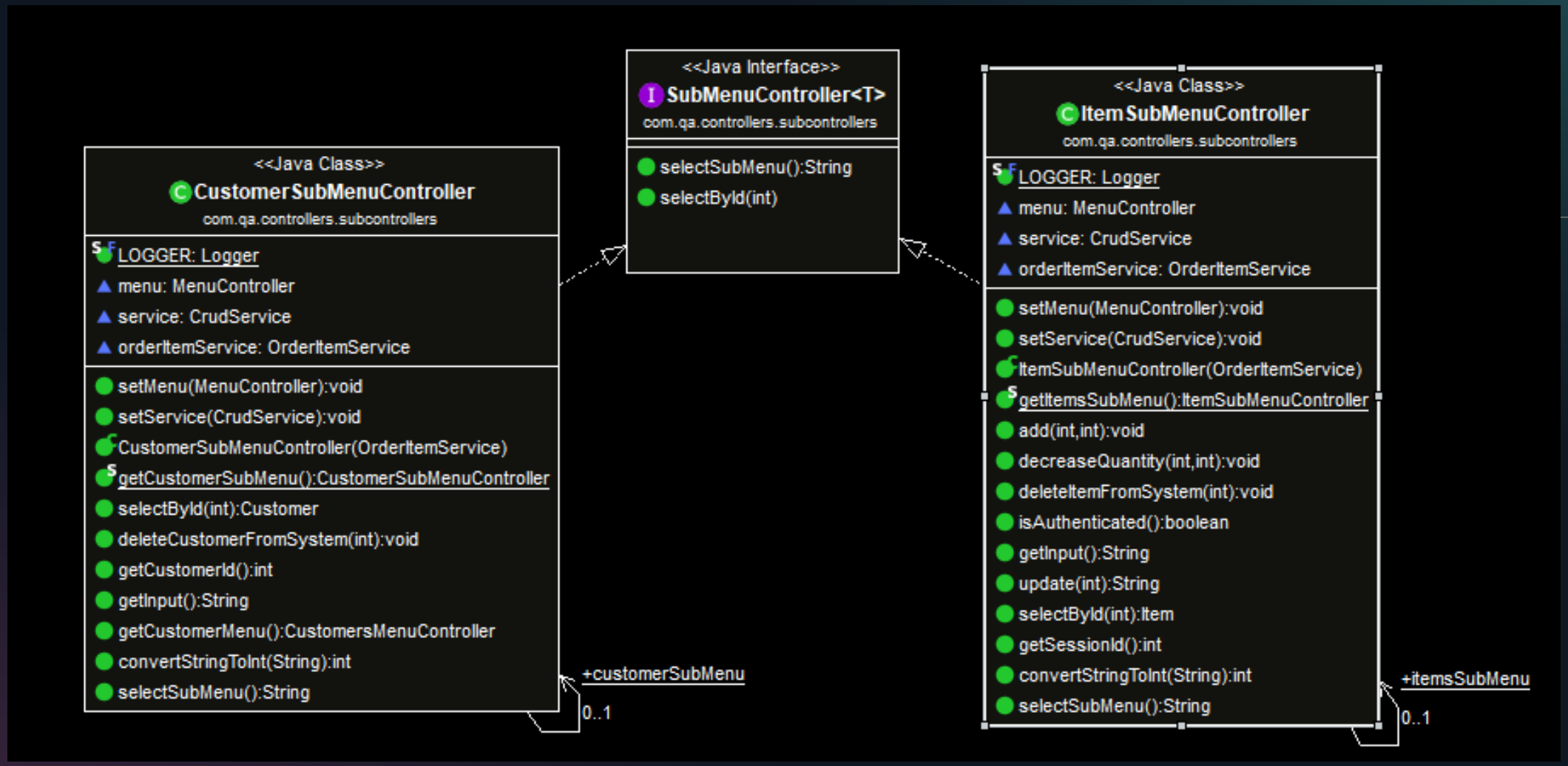

User
PK
User_ID
int
Username
varchar(30)
Password
varchar(40)
IsAdmin
boolean
Customer
PK
Customer_ID
int
FK
FK_User_ID
int
Name
varchar(40)
Order
PK
Order_ID
int
FK
FK_Customer_ID
int
Placed
date
Paid
boolean
Total
decimal(7,2)
Item
PK
Item_ID
int
Name
varchar(30)
Value
decimal(7,2)
Order_Item
PK
Order_Item_ID
int
FK
FK_Item_ID
int
FK
FK_Order_ID
int
Quantity
int

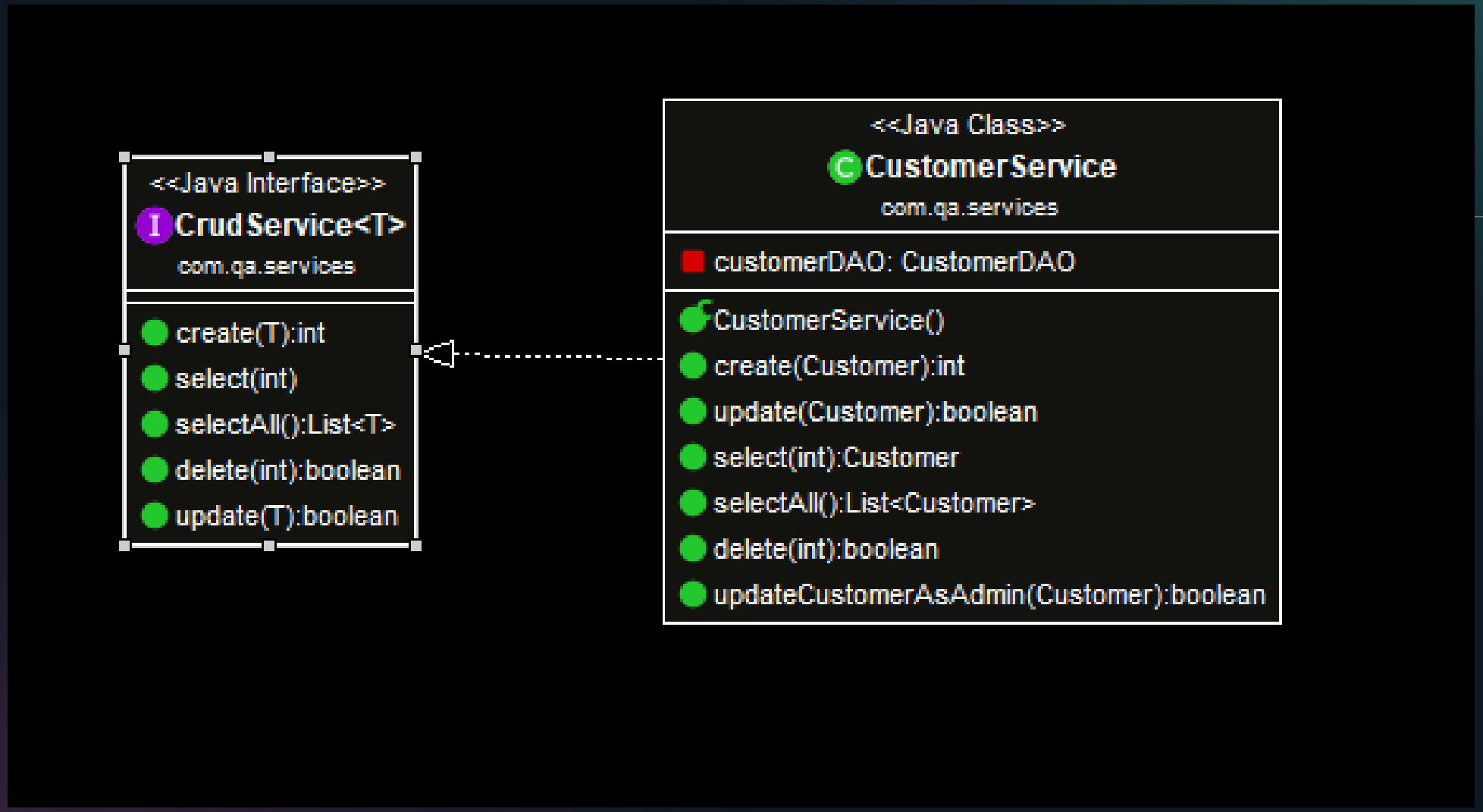
KANBAN BOARD BEFORE

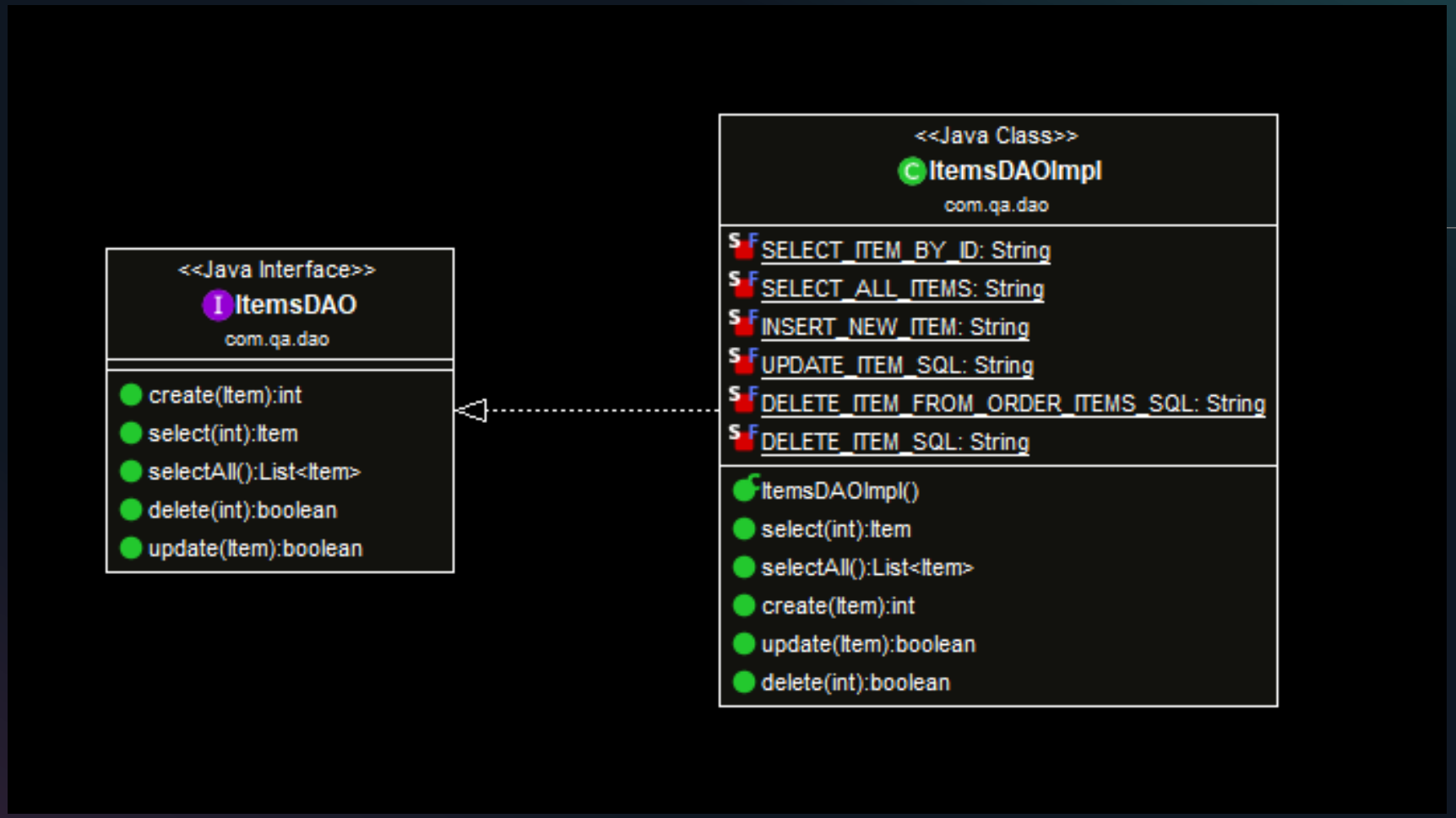



<<Java Class>>
Customer
com.qa.dto
id: Integer
name: String
userId: Integer
Customer(String)
Customer(Integer,String)
Customer(Integer,Integer)
getId():Integer
setId(Integer):void
getName():String
setName(String):void
getUserId():int
setUserId(Integer):void
equals(Object):boolean
-customer
0..1
<<Java Class>>
Order
com.qa.dto
id: Integer
customerId: Integer
placed: Date
paid: boolean
total: Double
Order()
Order(Integer)
Order(Integer,Customer,Date,boolean,Double)
getId():Integer
setId(Integer):void
getCustomerId():Integer
setCustomerId(Integer):void
getPlaced():Date
setPlaced(Date):void
isPaid():boolean
setPaid(boolean):void
getTotal():Double
setTotal(Double):void
getCustomer():Customer
setCustomer(Customer):void
equals(Object):boolean
<<Java Class>>
Item
com.qa.dto
id: Integer
name: String
value: Double
Item(Integer,String,Double)
Item(String,Double)
getId():Integer
setId(Integer):void
getName():String
setName(String):void
getValue():Double
setValue(Double):void
equals(Object):boolean
-item
0..1
<<Java Class>>
OrderItem
com.qa.dto
orderItemId: Integer
itemId: Integer
orderId: Integer
quantity: Integer
OrderItem(Item,Integer)
OrderItem(Integer,Integer)
getOrderItemId():Integer
setOrderItemId(Integer):void
getItemId():int
setItemId(Integer):void
getOrderId():int
setOrderId(Integer):void
getQuantity():Integer
setQuantity(Integer):void
getItem():Item
setItem(Item):void
equals(Object):boolean

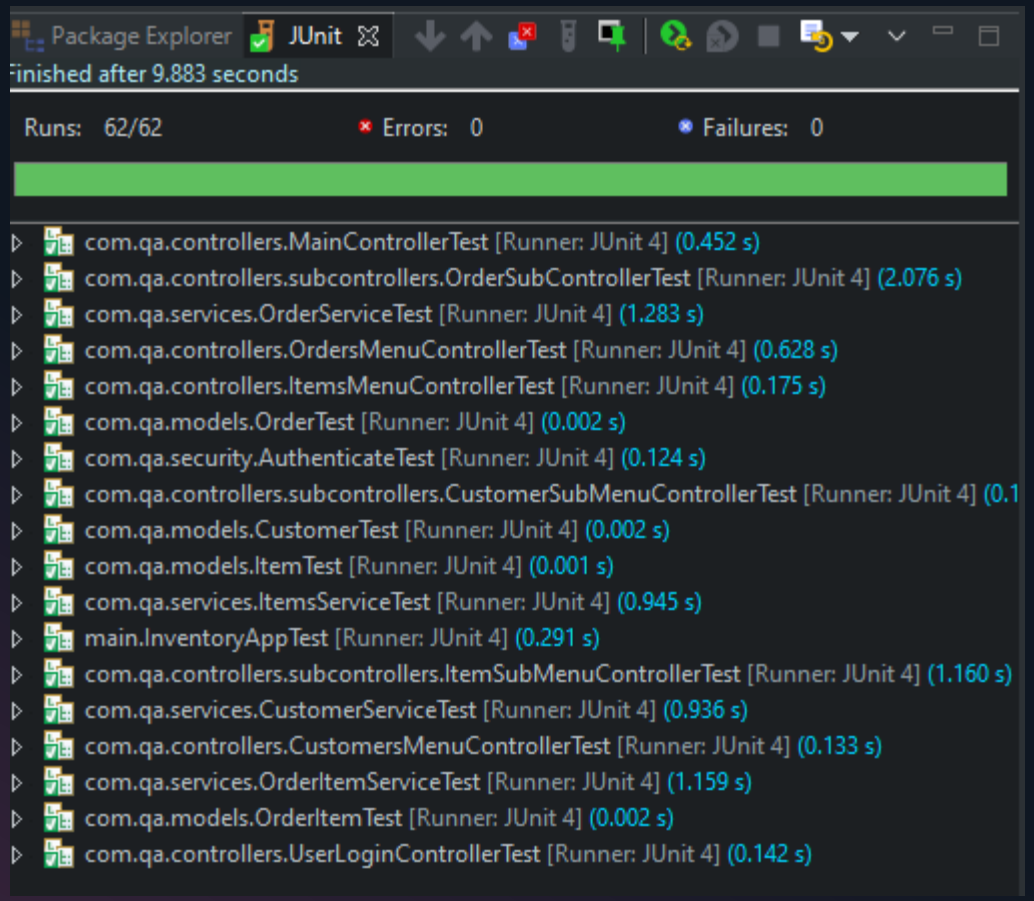

<<Java Interface>>
MenuController<T>
com.qa.controllers
selectMenuOptions():String
viewAll():List<T>
<<Java Class>>
CustomersMenuController
com.qa.controllers
LOGGER: Logger
subMenu: SubMenuController
service: CrudService
setMenu(MenuController):void
setSubMenu(SubMenuController):void
setService(CrudService):void
CustomersMenuController(CrudService)
CustomersMenuController()
getCustomerMenu():CustomersMenuController
viewAll():List<Customer>
getCustomerId():int
update():String
-menu
0..1
-menu
0..1
+customerMenu
0..1
<<Java Class>>
ItemsMenuController
com.qa.controllers
LOGGER: Logger
subMenu: SubMenuController
service: CrudService
setMenu(MenuController):void
setSubMenu(SubMenuController):void
setService(CrudService):void
ItemsMenuController(CrudService)
getItemsMenu():ItemsMenuController
viewAll():List<Item>
add():String
getInput():String
isAuthenticated():boolean
validateName(String):boolean
convertStringToDouble(String):double
+itemsMenu
0..1


<<Java Interface>>
SubMenuController<T>
com.qa.controllers.subcontrollers
selectSubMenu():String
selectById(int)
<<Java Class>>
CustomerSubMenuController
com.qa.controllers.subcontrollers
LOGGER: Logger
menu: MenuController
service: CrudService
orderItemService: OrderItemService
setMenu(MenuController):void
setService(CrudService):void
CustomerSubMenuController(OrderItemService)
getCustomerSubMenu():CustomerSubMenuController
selectById(int):Customer
deleteCustomerFromSystem(int):void
getCustomerId():int
getInput():String
getCustomerMenu():CustomersMenuController
convertStringToInt(String):int
selectSubMenu():String
+customerSubMenu
0..1
<<Java Class>>
ItemSubMenuController
com.qa.controllers.subcontrollers
LOGGER: Logger
menu: MenuController
service: CrudService
orderItemService: OrderItemService
setMenu(MenuController):void
setService(CrudService):void
ItemSubMenuController(OrderItemService)
getItemsSubMenu():ItemSubMenuController
add(int,int):void
decreaseQuantity(int,int):void
deleteItemFromSystem(int):void
isAuthenticated():boolean
getInput():String
update(int):String
selectById(int):Item
getSessionId():int
convertStringToInt(String):int
selectSubMenu():String
+itemsSubMenu
0..1


<<Java Interface>>
CrudService<T>
com.qa.services
create(T):int
select(int)
selectAll():List<T>
delete(int):boolean
update(T):boolean
<<Java Class>>
CustomerService
com.qa.services
customerDAO: CustomerDAO
CustomerService()
create(Customer):int
update(Customer):boolean
select(int):Customer
selectAll():List<Customer>
delete(int):boolean
updateCustomerAsAdmin(Customer):boolean

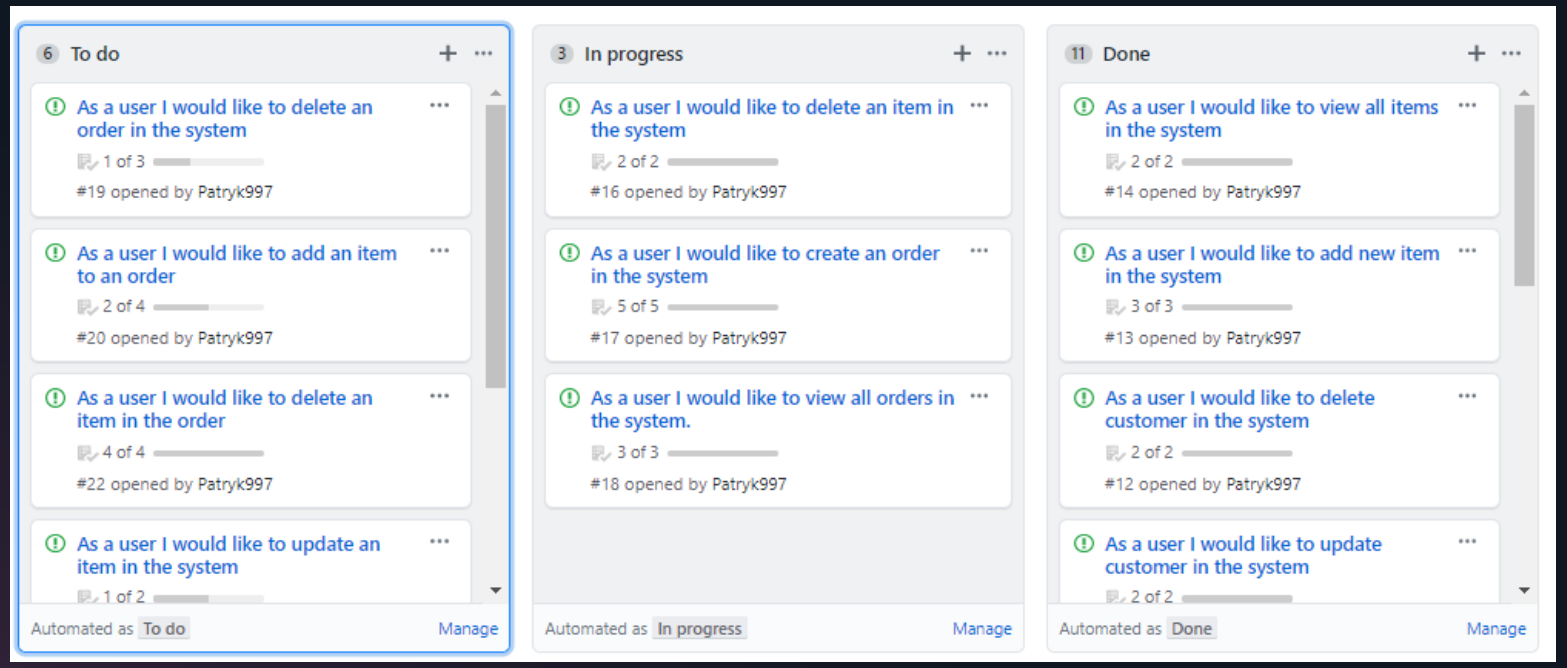

<<Java Interface>>
ItemsDAO
com.qa.dao
create(Item):int
select(int):Item
selectAll():List<Item>
delete(int):boolean
update(Item):boolean
<<Java Class>>
ItemsDAOImpl
com.qa.dao
SELECT_ITEM_BY_ID: String
SELECT_ALL_ITEMS: String
INSERT_NEW_ITEM: String
UPDATE_ITEM_SQL: String
DELETE_ITEM_FROM_ORDER_ITEMS_SQL: String
DELETE_ITEM_SQL: String
ItemsDAOImpl()
select(int):Item
selectAll():List<Item>
create(Item):int
update(Item):boolean
delete(int):boolean

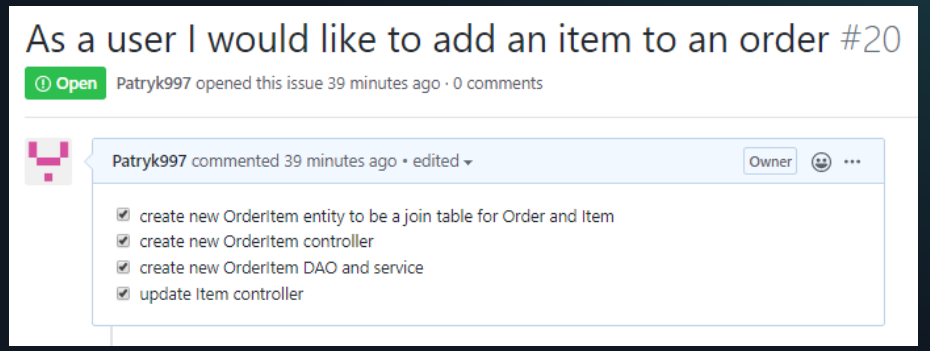
TEST COVERAGE



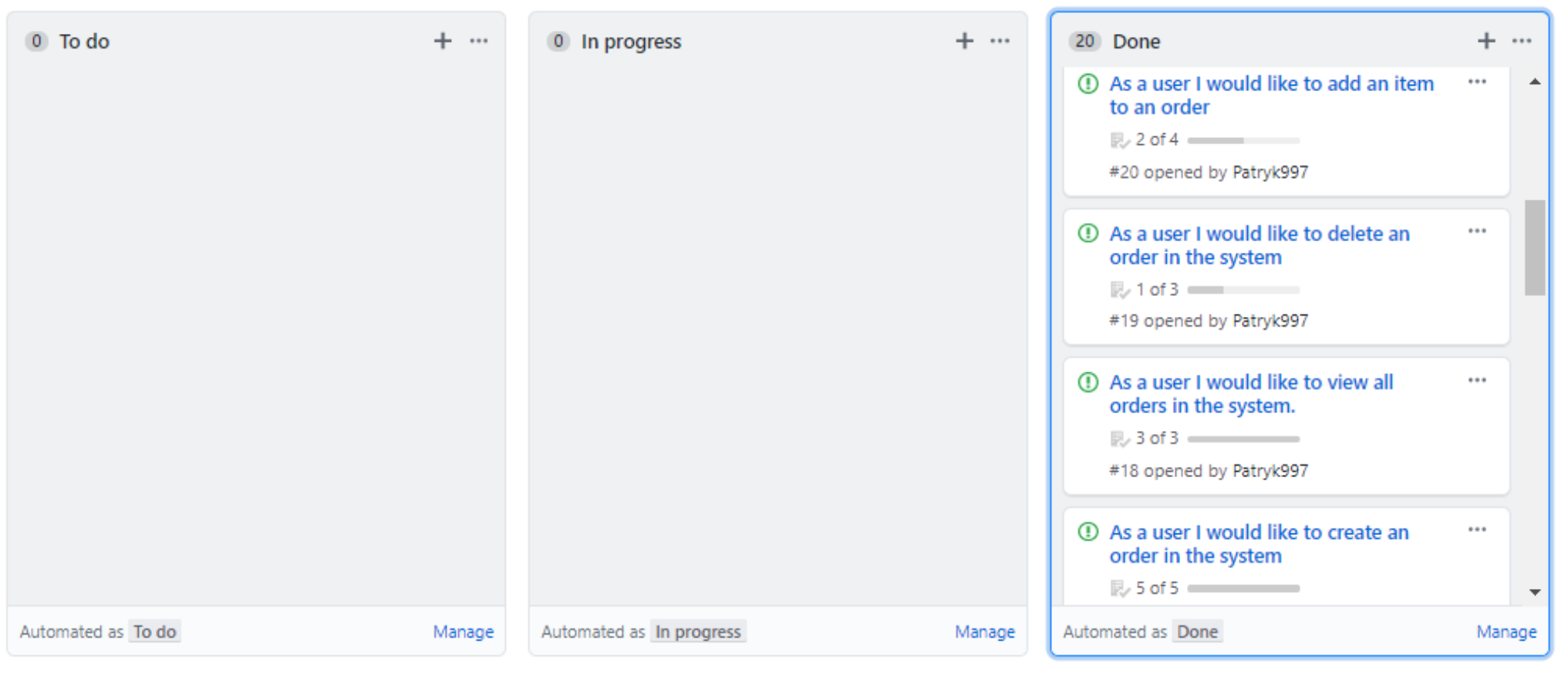
Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
QA-Java	81.7 %	5,553	1,242	6,795

KANBAN BOARD IN THE MIDDLE OF THE PROJECT

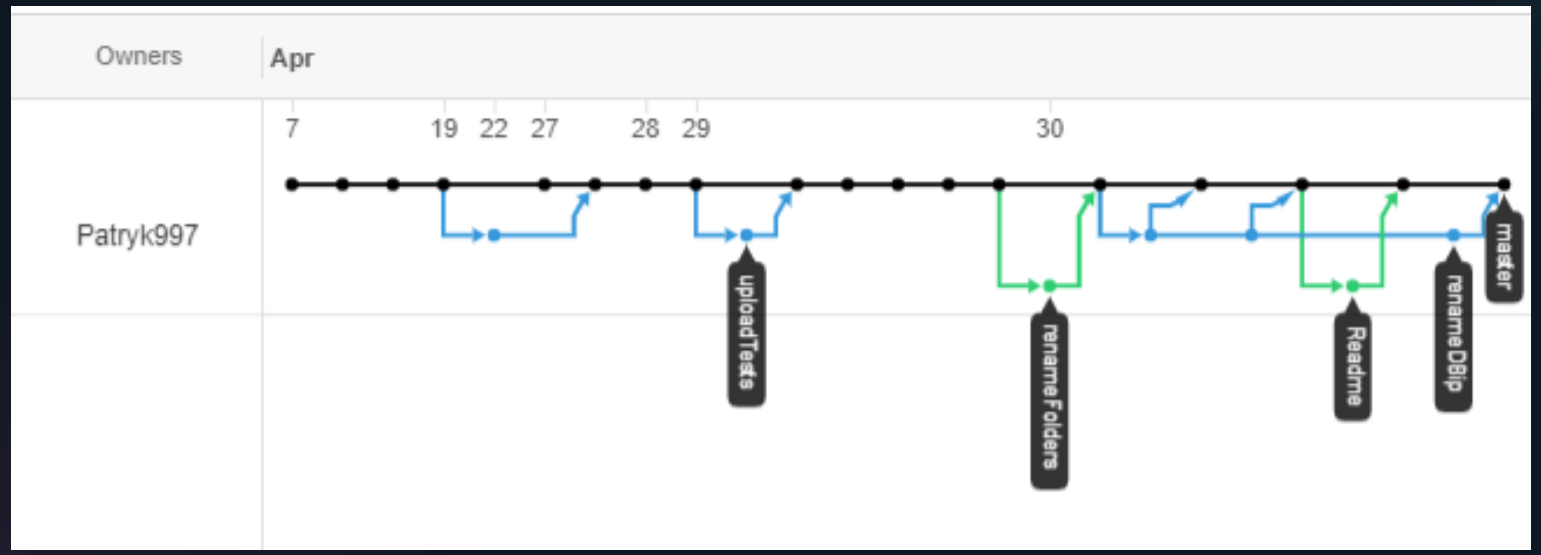




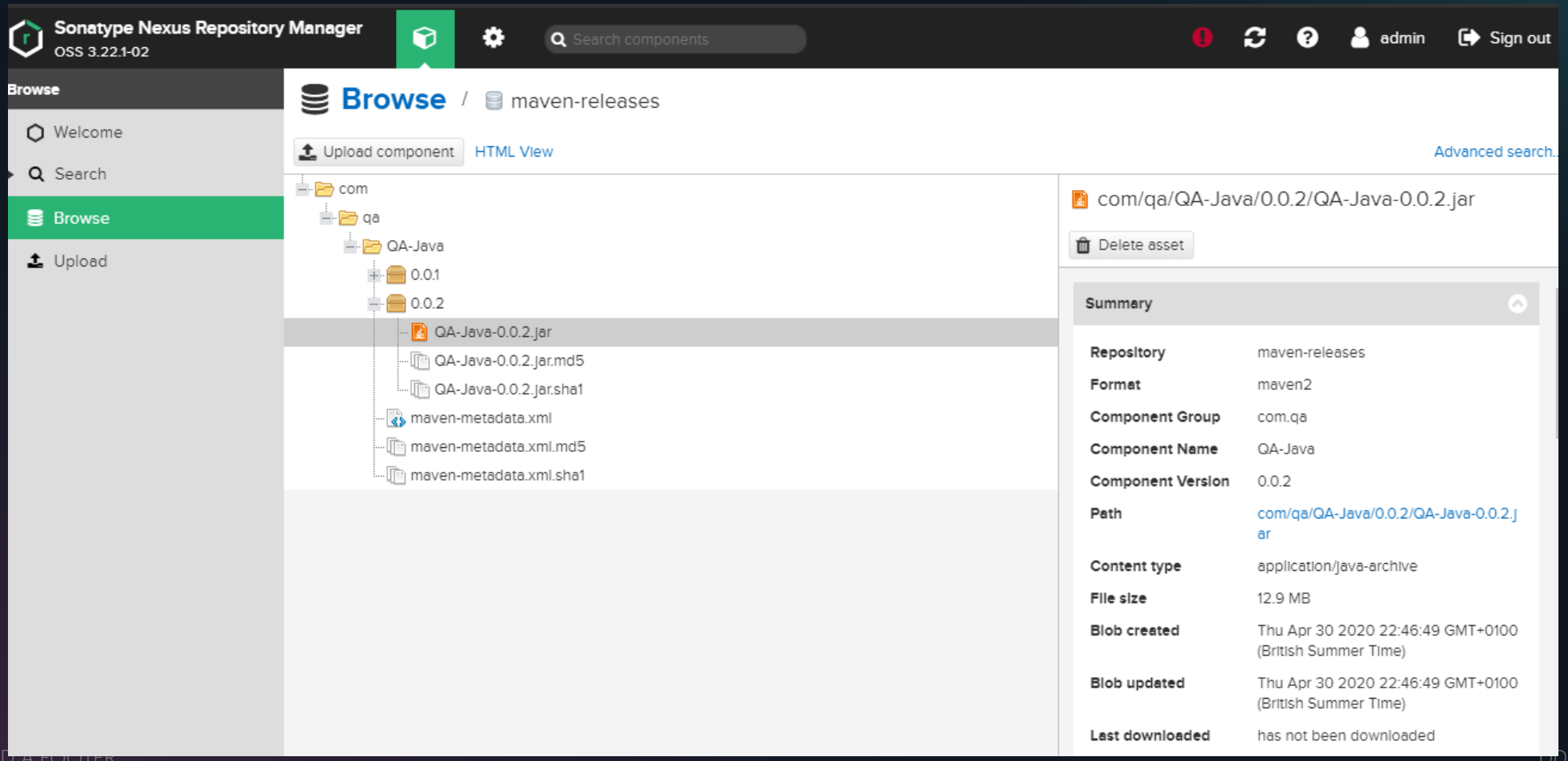
KANBAN BOARD AFTER



GITHUB NETWORK COMMITS



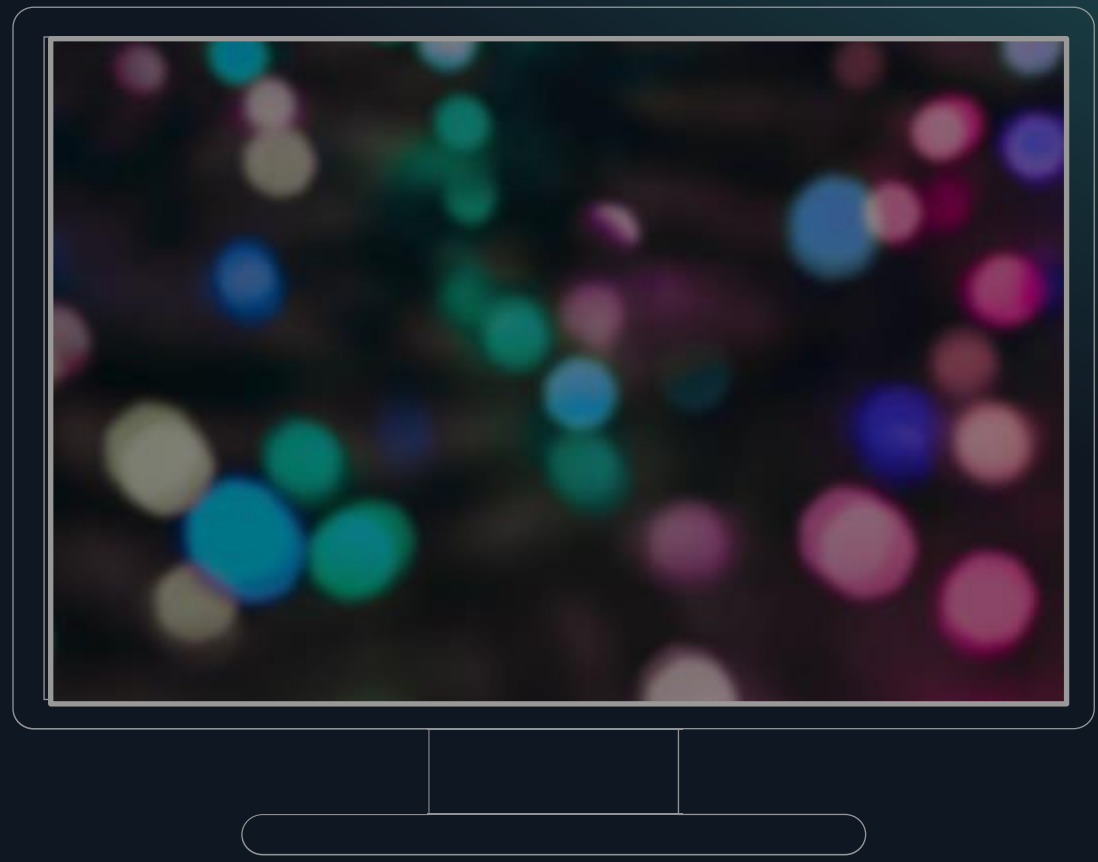
NEXUS REPOSITORY



RETROSPECTIVE

WHAT WAS DONE WELL, WHAT TO IMPROVE

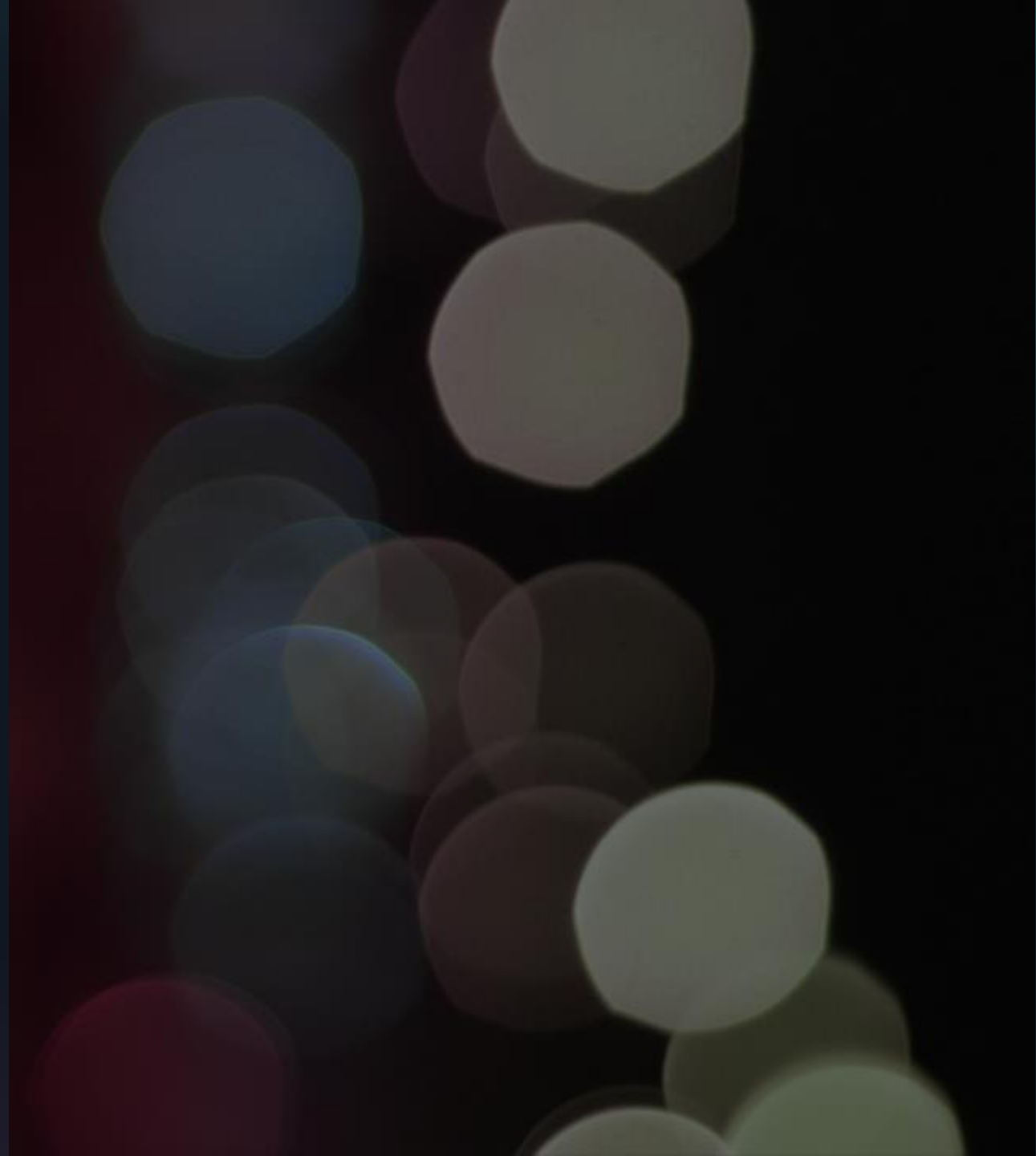
- Project status urgent should be given from day 1
- Should follow more OOP and SOLID principles
- Followed 3 Layer application architecture



THANK YOU!

Patryk Maryn

Phone number:
07908 658 465

Email address:
patmaryn31@gmail.com

Website:
https://www.patrykmaryn.co.uk