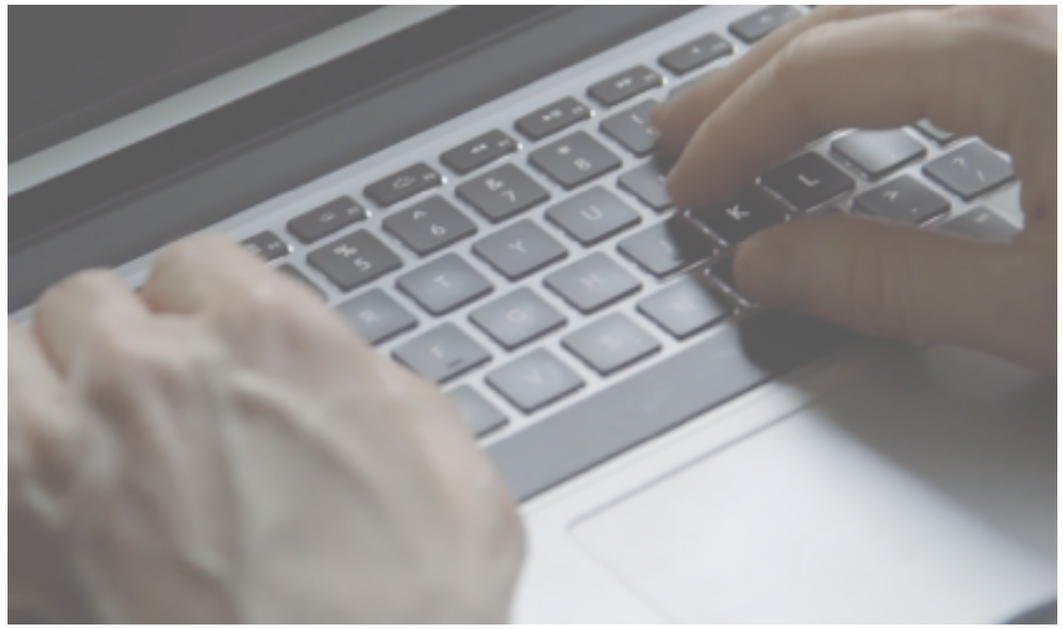
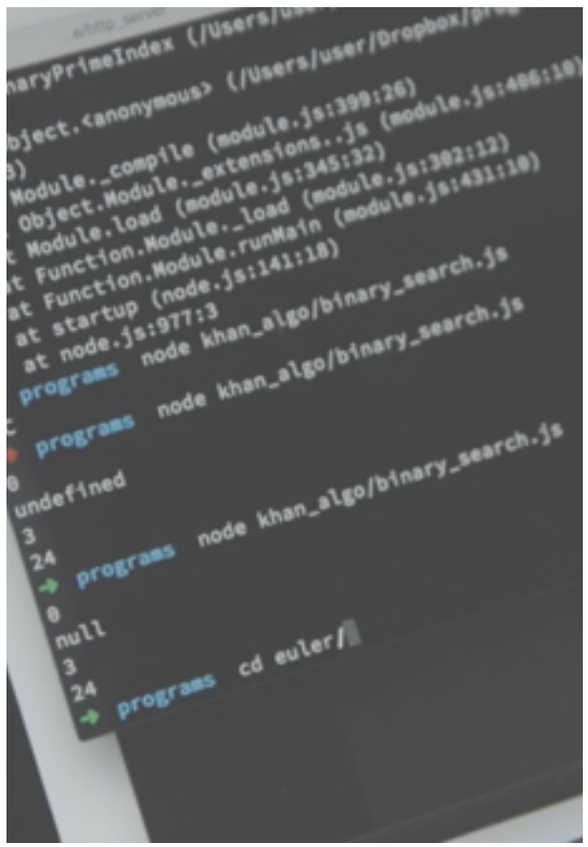
Final Evidence to Gather for the PDA





Learning Objectives

Aim: To help in the completion of evidence from a software development project for the PDA.

- Know what is required for the final PDA unit.
- Understand how to complete the various diagrams.
- Be able to produce evidence that can be submitted as part of the final unit of the PDA.

Acceptance Criteria

An Acceptance Criteria is a set of statements with a clear pass/fail result. Acceptance Criteria should state intent, but not a solution.

Acceptance Criteria	Expected Result/Output	Pass / Fail
A user is able to access a list of available reports.		
A manager can approve or disapprove an audit form		

Acceptance Test Plan

An Acceptance Test Plan is the final two columns of this table.

Acceptance Criteria	Expected Result/Output	Pass / Fail
A user is able to access a list of available reports.	List of available reports is displayed when the list link is clicked.	Pass
A manager can approve or disapprove an audit form	When manager clicks on approve report button, page is displayed of reports with pop-up confirming report approved.	Pass

System Interaction Diagrams

System interaction diagrams look at the flow of control and data among the things in the system.

There are two types of system interaction diagrams:

- **Sequence Diagrams**

More information at http://www.ibm.com/developerworks/rational/library/3101.html

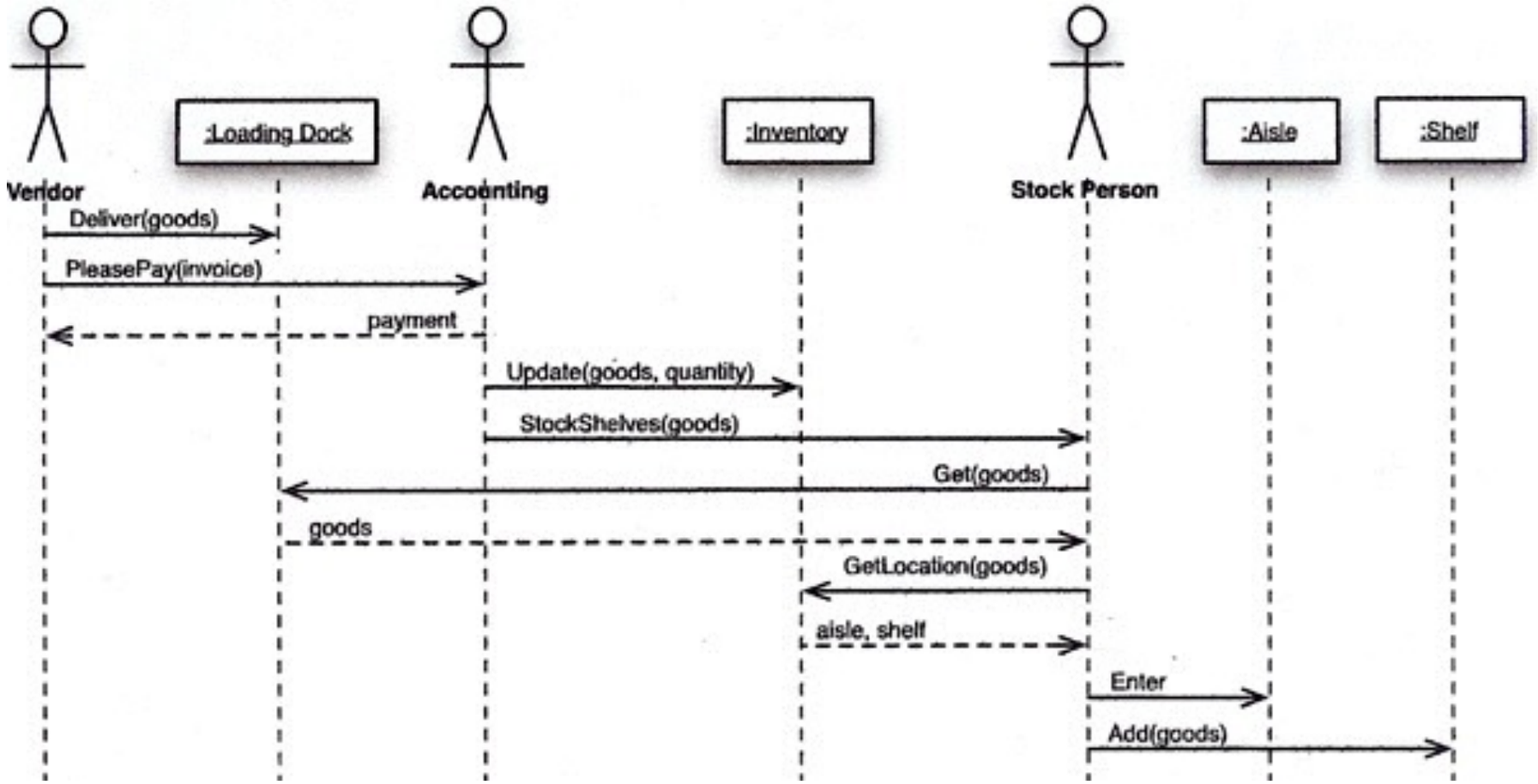
- **Collaboration Diagrams**

More information at http://www.ibm.com/developerworks/rational/library/3101.html

Sequence Diagrams

The behaviour of objects in a use case by describing the objects and the messages they pass.

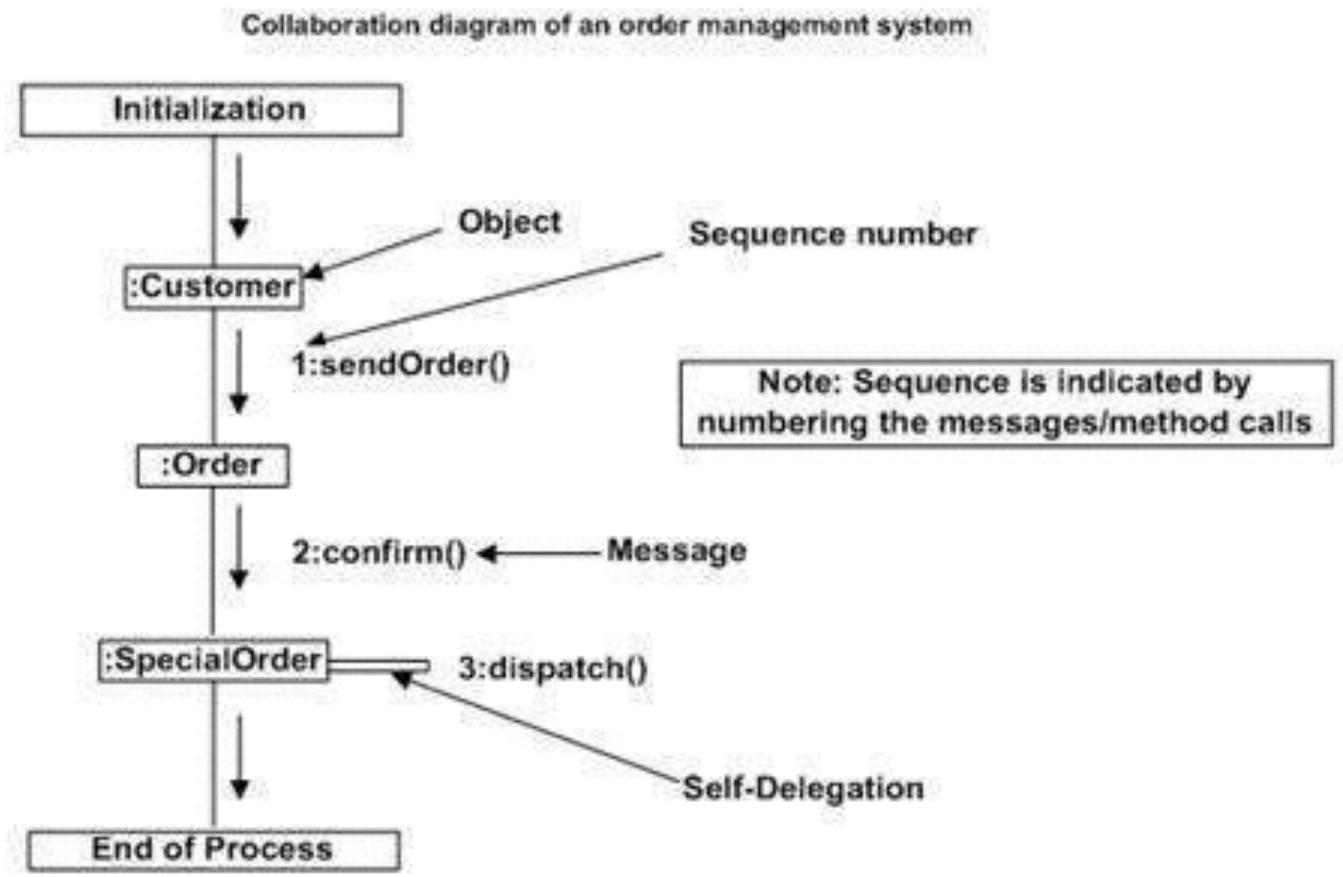
- Order in which messages occur
- What messages are sent between a system's objects



Collaboration Diagrams

Collaboration diagrams show the structural organization of objects taking part in the interaction.

- Relationship between objects
- The order of messages passed between them



Bug Tracking Report

A Bug Tracking Report is a record of all the bugs that occur in the process of building a Software Product.

For the PDA you must include 5-6 examples of “bugs” or errors.

User must be able to add a trip	Failed	Saving a user, using the ID to assign a trip	Passed
Trip has a starting and end date			Passed
Trip date cannot be made for dates passed	Failed	Added validations to stop creation of trips with past dates	Passed
Trip can only have a number of available spaces	Failed	Set a number of spaces available per trip.	Passed

Testing Evidence

Screenshots of test code and tests being run

Minimum of three tests

Must show failures as well as passing

```
game_homework_lab  mocha specs


food
  ✓ should create a piece of food with a name
  ✓ should create a piece of food with a replenishment value

hero
  ✓ should create a hero with a name
  ✓ should create a Hero with health out of 20
  ✓ should create a Hero with a favourite food
  ✓ should create a Hero with a weapon
  ✓ should be able to talk
  ✓ should have the health of the Hero go up when they eat food
  ✓ should have the health of the Hero go up by 1.5 times when they eat their favourite food
  ✓ should have health go down when Hero eats poisoned food
  ✓ should be able to attack with bow and arrow
  ✓ should be able to attack with sword
  ✓ should be able to attack with staff
  ✓ should block the villains attack if heroes health is more than 5

rat
  ✓ should create a rat with a name
  ✓ should make food poisonous when touched

Villain
  ✓ should create a Villain with a name
  ✓ should create a Villain with an age
  ✓ should create a Villain with health that equals
  ✓ should create a Villain with a weapon
  ✓ should have an intial empty array of minions
  ✓ should add a rat to the array of minions
  ✓ should create an army of rats
  ✓ should damage the hero with a curse
  ✓ should damage the hero with an attack

25 passing (21ms)
```

```
it('should create a Hero with a weapon', function(){
  var sword = new Weapon("Sword", 5);
  var aragorn = new Hero("Aragorn", 20, "Apples", sword);
  assert.equal("Sword", aragorn.heroWeapon.type);
});

it('should be able to talk', function(){
  var legolas = new Hero("Legolas", 20, "Lembas bread");
  legolas.talk();
  assert.equal("Hello, my name is Legolas", legolas.talk());
});

it('should have the health of the Hero go up when they eat food'
  , function(){
  var legolas = new Hero("Legolas", 20, "Lembas bread");
  var apple = new Food("Apples", 5);
  legolas.eat(apple);
  assert.equal(25, legolas.health);
});
```

What happens now?

In your PDA repo you should have the following things saved:

- Analysis & Design Unit Evidence Document
- Implementation & Testing Unit Evidence Document
- Project Unit Evidence Document
- A completed Static and Dynamic Coding Exercise (Tasks A)
- A completed Unit and Integration Coding Exercise (Tasks B)

Deadline: Two weeks after the course finishes