

**PDA: Software Development
Level 8
Student Evidence Checklist**

Full name	Christopher Murphy
Cohort	G4

The evidence required can be taken from your assignments, homework that you have completed on your own or by creating a specific example for the PDA.

	Unit	Ref.	Evidence	Done
	I & T	I.T 5	Demonstrate the use of an array in a program. Take screenshots of: *An array in a program *A function that uses the array *The result of the function running	
			<pre>const RecordCollector = function(name,cash){ this.name = name; this.collection = [] this.cash = cash; }</pre>	
Week 2			<pre>RecordCollector.prototype.buy = function (record) { if(this.cash > record.price){ this.cash -= record.price; this.collection.push(record) } };</pre>	

		<pre>it('record collector can buy a record', function(){ recordCollector.buy(record1); assert.strictEqual(recordCollector.cash, 61); assert.deepStrictEqual(recordCollector.collection, [record1]); })</pre>	<ul style="list-style-type: none"> ✓ record store is initially empty ✓ can add records ✓ can show store balance ✓ can print records details ✓ store can list its inventory ✓ store can sell a record ✓ store can show finances ✓ store can view records by genre ✓ record collector can buy a record 	
I & T	I.T 6			
I & T		Static and Dynamic testing task A		

Unit	Ref.	Evidence	Done
I & T	I.T 3	Demonstrate searching data in a program. Take screenshots of: *Function that searches data *The result of the function running	

Week 3

```
public static <T> List<T> getAll(Class classType){
    session = HibernateUtil.getSessionFactory().openSession();
    List<T> results = null;
    Criteria criteria = session.createCriteria(classType);
    results = getList(criteria);
    return results;
}
```

```
List<Team> foundTeams = DBHelper.getAll(Team.class);
```

```
foundTeams = {ArrayList@2167} size = 5
  0 = {Team@2175}
  1 = {Team@2176}
  2 = {Team@2177}
  3 = {Team@2178}
  4 = {Team@2179}
```

I & T

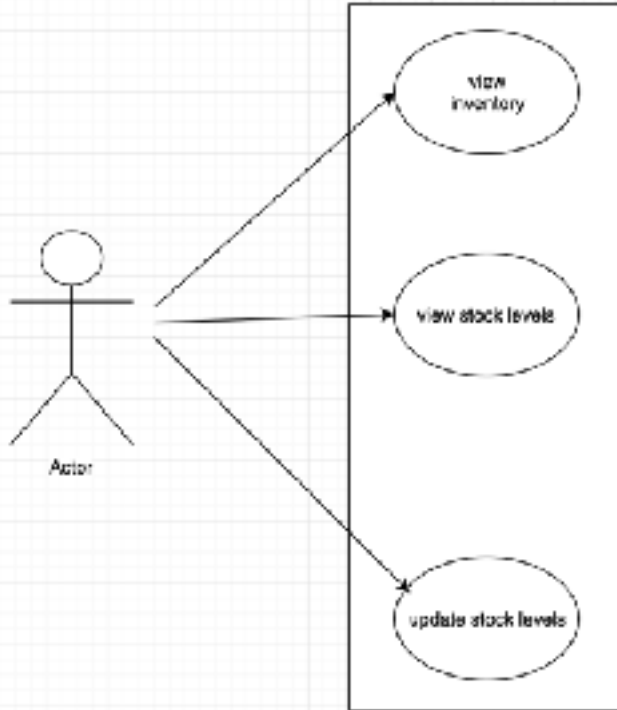
I.T 4

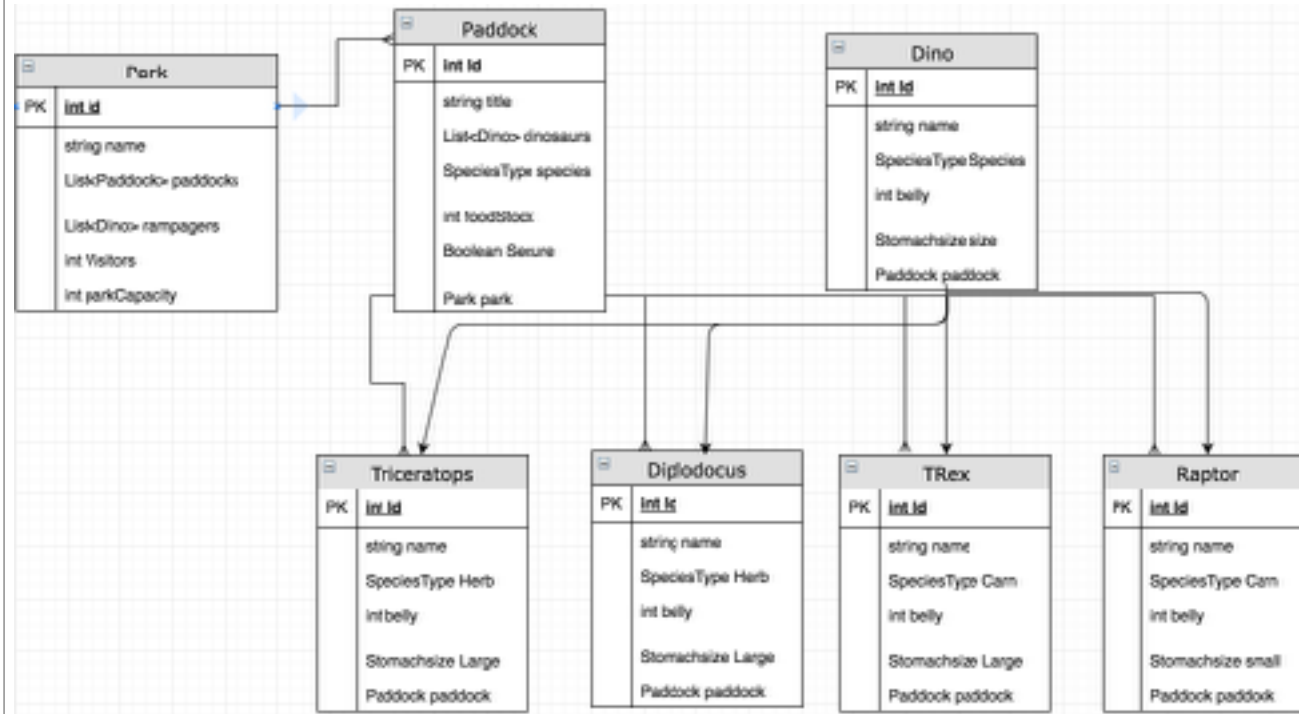
Demonstrate sorting data in a program. Take screenshots of:
 *Function that sorts data
 *The result of the function running

```
public static List<Team> getTeamsInLeague(League league){  
    session = HibernateUtil.getSessionFactory().openSession();  
    List<Team> results = null;  
    Criteria criteria = session.createCriteria(Team.class);  
    criteria.add(Restrictions.eq("league", league));  
    criteria.addOrder(Order.desc("points"));  
    results = getList(criteria);  
    return results;  
}
```

```
List<Team> teamsfoundInPointsOrder = DBHelper.getTeamsInLeague(league);
```

```
▼ teamsfoundInPointsOrder = {ArrayList@2170} size = 5
  ▼ 0 = {Team@2183}
    id = 5
    name = "Barcelona"
    points = 20
    manager = null
    league = {League@2189}
  ▼ 1 = {Team@2184}
    id = 4
    name = "Newcastle"
    points = 19
    manager = null
    league = {League@2189}
  ▼ 2 = {Team@2185}
    id = 1
    name = "soccer united"
    points = 12
    manager = null
    league = {League@2189}
  ▼ 3 = {Team@2186}
    id = 3
    name = "Man blues"
    points = 7
    manager = null
    league = {League@2189}
  ▼ 4 = {Team@2187}
    id = 2
    name = "Man reds"
    points = 3
    manager = null
    league = {League@2189}
```

Unit	Ref.	Evidence	Done
A & D	A.D 1	A Use Case Diagram	
		 <pre> graph LR Actor((Actor)) --> UC1((view inventory)) Actor --> UC2((view stock levels)) Actor --> UC3((update stock levels)) </pre> <p>The diagram is a Use Case Diagram. It features a stick figure actor on the left labeled 'Actor'. To the right of the actor is a large rectangular box containing three ovals, each representing a use case. The top oval is labeled 'view inventory', the middle one 'view stock levels', and the bottom one 'update stock levels'. Three arrows originate from the actor's right side and point to the center of each of the three use case ovals.</p>	
A & D	A.D 2	A Class diagram.	

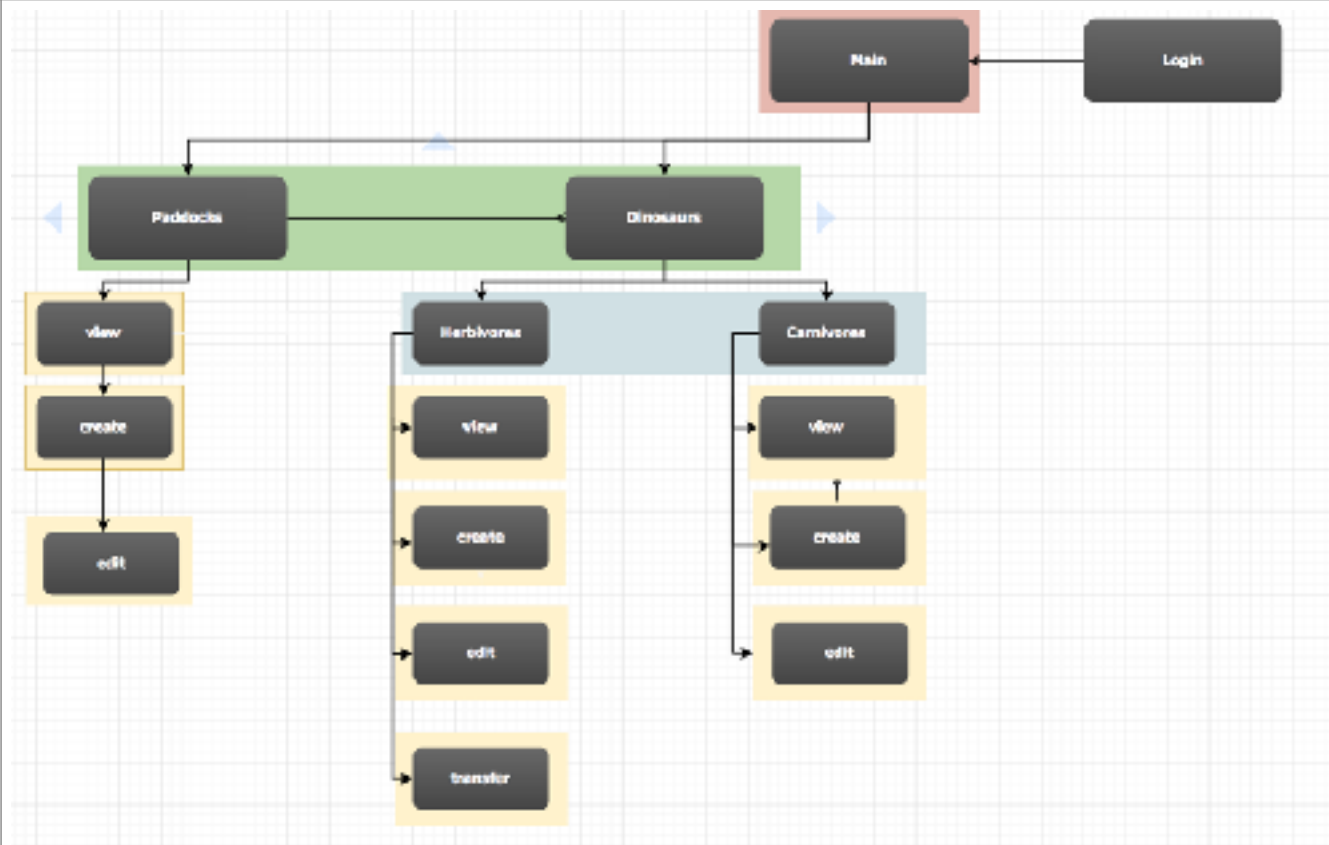


A & D A.D 3 An Object diagram.

A & D A.D 4 An Activity Diagram

A & D A.D 6 Produce an Implementations Constraints plan detailing the following factors:
 *Hardware and software platforms
 *Performance requirements
 *Persistent storage and transactions
 *Usability
 *Budgets
 *Time

P P 5 Create a user sitemap.



P

P 6

Produce two wireframe designs.

Paddocks

paddock	Type	Status	view	update
Paddock 1	Herbivore	Food Stock Low!	view	update
paddock 2	Carnivore	Safe	view	update
paddock 3	Carnivore	No Stock! Re-stock immediately	view	update

[Add Paddock](#)

Week 5



P

P 10

Take a screenshot of an example of pseudocode for a function.

```

let newArray = [];
// for each index position in arr.
for(let i in arr){
    // if the index number is less than the number passed in.
    if(1 < index){
        // push those elements into the newArray.
        newArray.push(arr[i]);
    }else{
        // then push itemToAdd to the end of the newArray.
        newArray.push(itemToAdd);
        //then push the remaining elements to the end of newArray.
        newArray.push(arr[i]);
    }
}

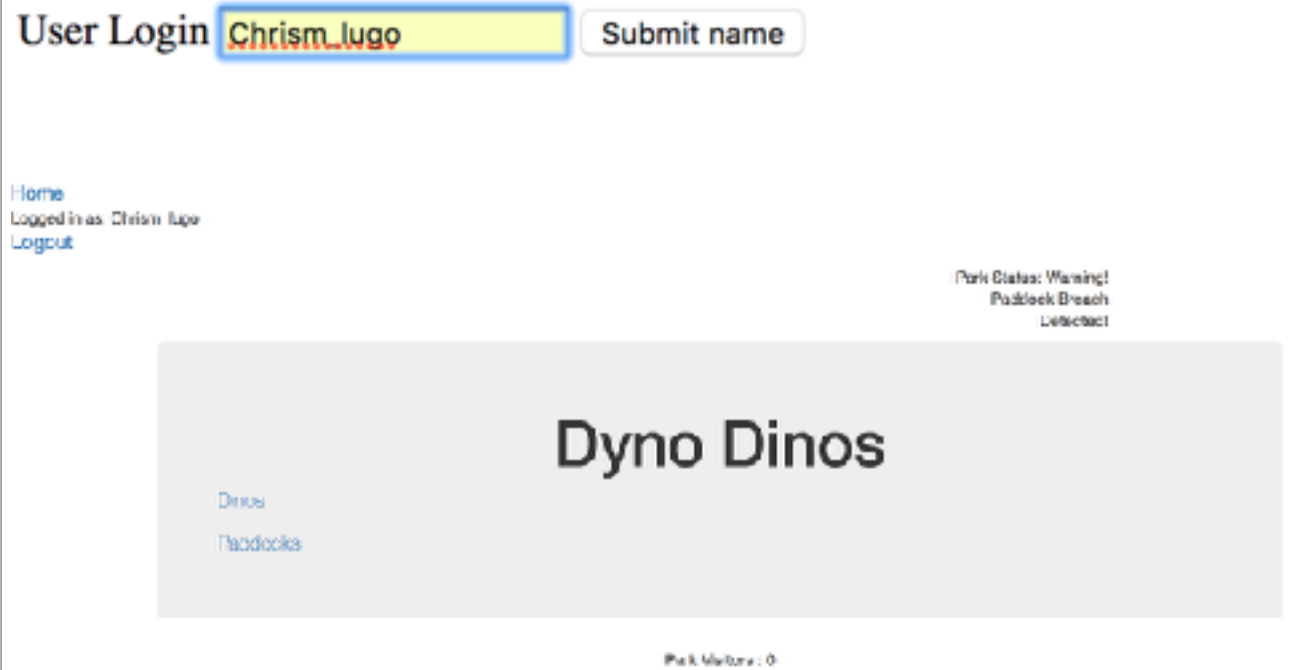
```

P P 13 Show user input being processed according to design requirements. Take a screenshot of:

- * The user inputting something into your program
- * The user input being saved or used in some way

P P 14 Show an interaction with data persistence. Take a screenshot of:

- * Data being inputted into your program
- * Confirmation of the data being saved

			
P	P 15	<p>Show the correct output of results and feedback to user. Take a screenshot of:</p> <ul style="list-style-type: none"> * The user requesting information or an action to be performed * The user request being processed correctly and demonstrated in the program 	

Home
Logged in as: Chrisn_jago
Logout

All Paddocks

New Paddock

Name	Species	Food Stock	Status	View	edit	Delete
Herbs Green	HERBIVORE	0	True	View	edit	Delete
Herb Bush	CARNIVORE	0	False	View	edit	Delete
Raptors Paradise	CARNIVORE	0	False	View	edit	Delete

Park Visitors : 0

Downloaded paddocks

Home
Logged in as: Chrisn_jago
Logout

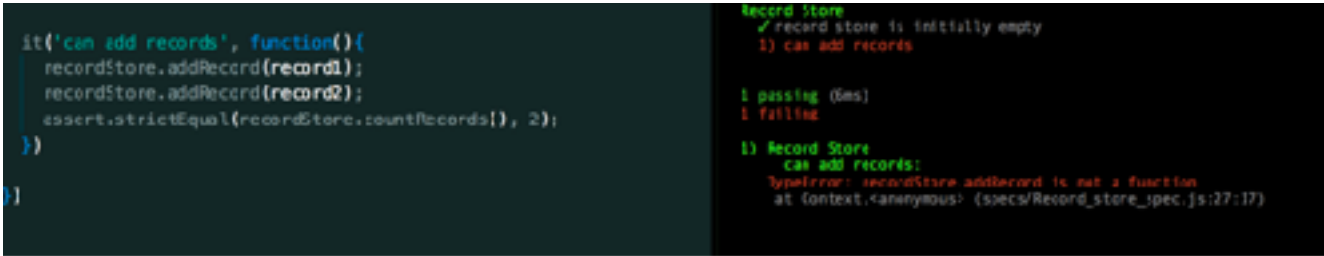
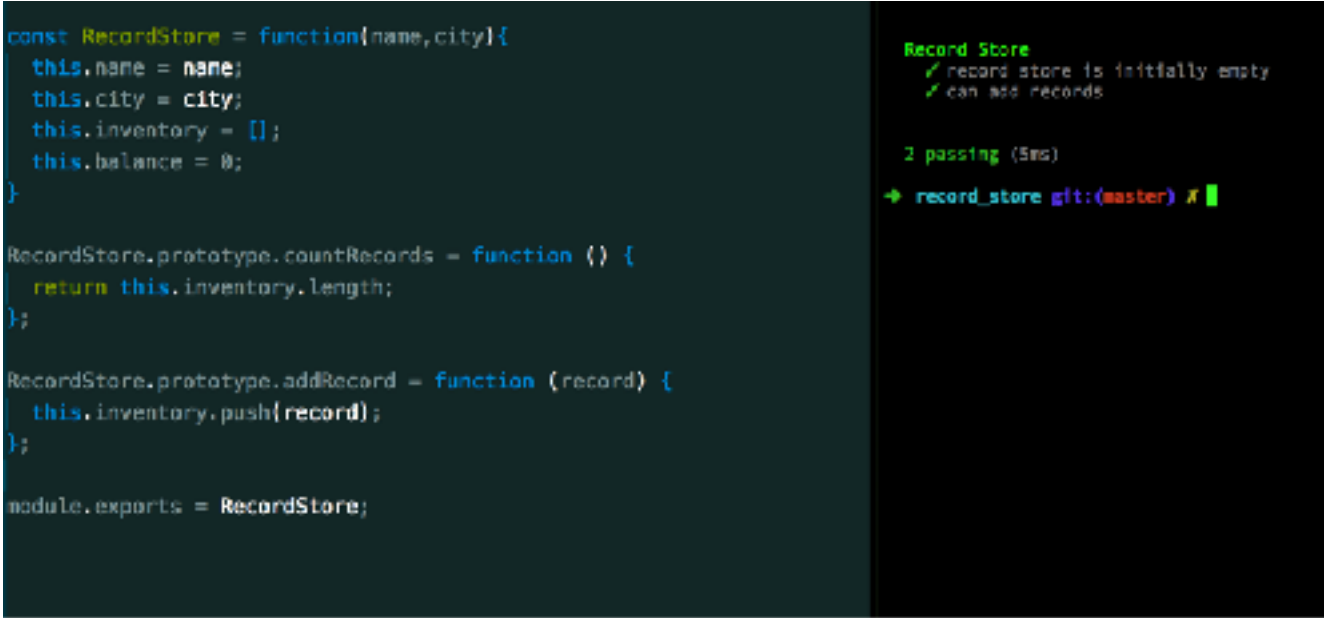

Paddock: Herbs Green

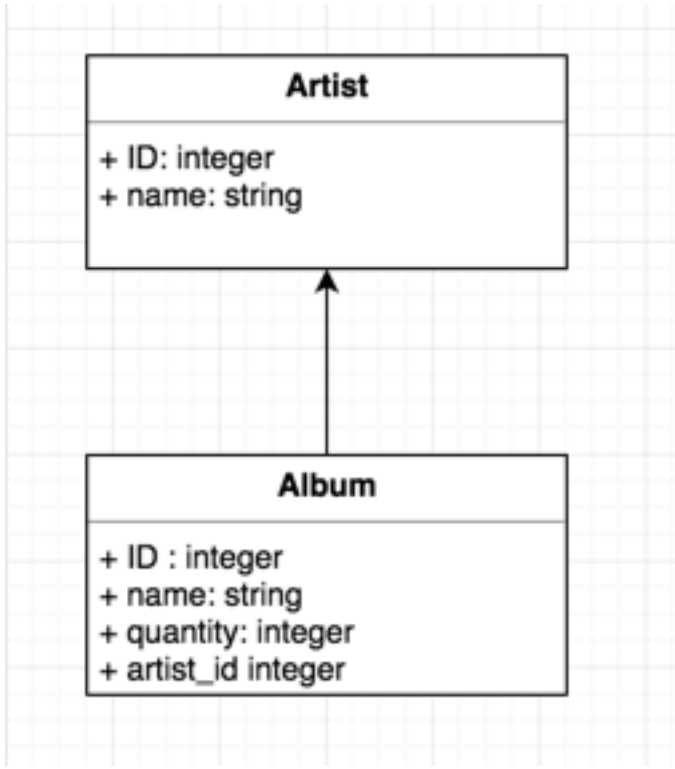
Re-Stock Paddock

Dinos

Name	Species	Status
LittleFoot	HERBIVORE	Unhappy: Check Herbs Green food store or check for paddock transfer
Cora	HERBIVORE	Unhappy: Check Herbs Green food store or check for paddock transfer
Woog	HERBIVORE	Unhappy: Check Herbs Green food store or check for paddock transfer

Park Visitors : 0

P	P 18	<p>Demonstrate testing in your program. Take screenshots of:</p> <ul style="list-style-type: none"> * Example of test code * The test code failing to pass * Example of the test code once errors have been corrected * The test code passing 	
		 <pre> it('can add records', function(){ recordStore.addRecord(record1); recordStore.addRecord(record2); assert.strictEqual(recordStore.countRecords(), 2); }) }) Record Store ✓ record store is initially empty 1) can add records 1 passing (6ms) 1 failing 1) Record Store can add records: TypeError: recordStore.addRecord is not a function at (context.<anonymous> (specs/Record_store_spec.js:27:17)) </pre>  <pre> const RecordStore = function(name,city){ this.name = name; this.city = city; this.inventory = []; this.balance = 0; } RecordStore.prototype.countRecords = function () { return this.inventory.length; }; RecordStore.prototype.addRecord = function (record) { this.inventory.push(record); }; module.exports = RecordStore; </pre>  <pre> Record Store ✓ record store is initially empty ✓ can add records 2 passing (5ms) → record_store gilt:(master) ✓ </pre>	

Unit	Ref.	Evidence	Done
I & T	I.T 7	Demonstrate the use of Polymorphism in a program.	
A & D	A.D 5	An Inheritance Diagram	
		 <pre> classDiagram class Artist { + ID: integer + name: string } class Album { + ID : integer + name: string + quantity: integer + artist_id integer } Artist < -- Album </pre>	
I & T	I.T 1	Take a screenshot of an example of encapsulation in a program.	
I & T	I.T 2	Take a screenshot of the use of Inheritance in a program. Take screenshots of: <ul style="list-style-type: none"> *A Class *A Class that inherits from the previous class *An Object in the inherited class *A Method that uses the information inherited from another class. 	

Week 7

```
public abstract class Employee {
    String name;
    String niNumber;
    double salary;

    public Employee(String name, String niNumber, double salary) {
        this.name = name;
        this.niNumber = niNumber;
        this.salary = salary;
    }
}
```

```
public class Manager extends Employee {
    private String deptName;

    public Manager(String name, String niNumber, double salary, String deptName) {
        super(name, niNumber, salary);
        this.deptName = deptName;
    }

    public String getDeptName() {
        return deptName;
    }
}
```

```
public class Director extends Manager {
    private double budget;

    public Director(String name, String niNumber, double salary, String deptName, double budget) {
        super(name, niNumber, salary, deptName);
        this.budget = budget;
    }

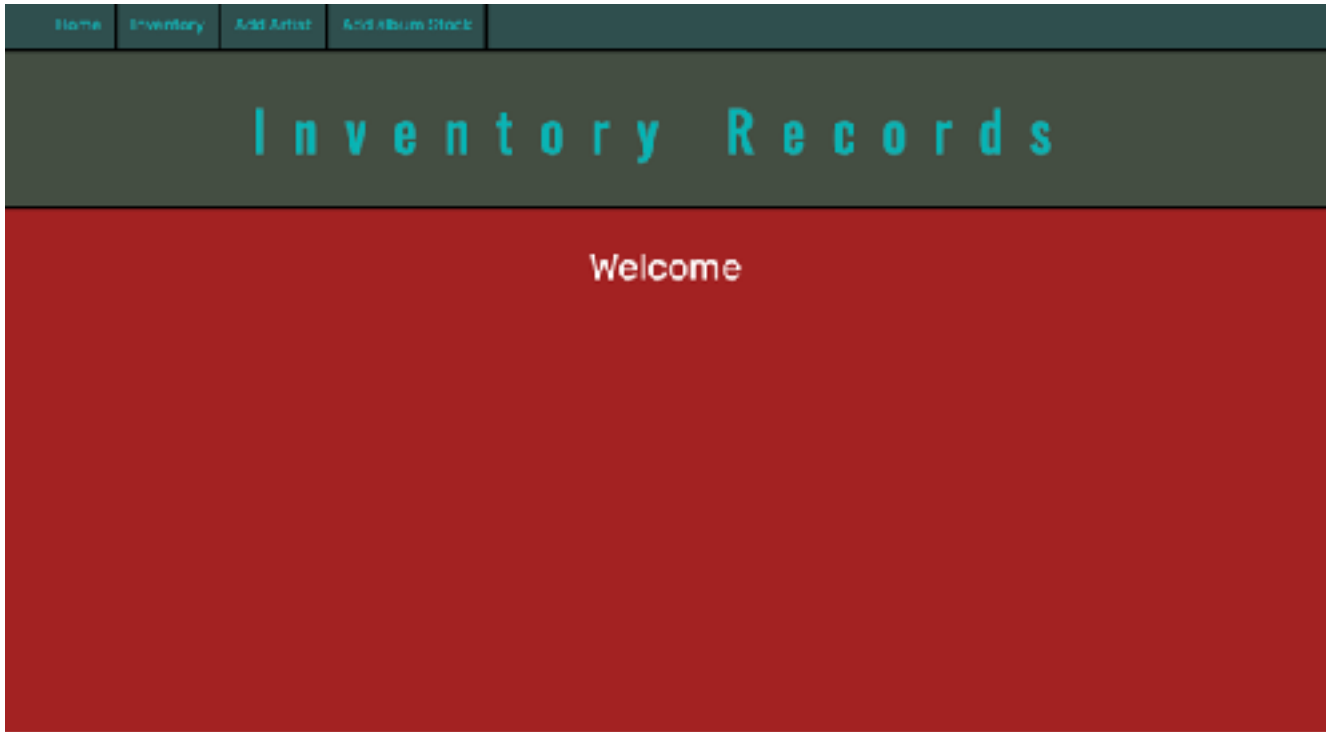
    public double getBudget() {
        return budget;
    }

    public double payBonus(){
        return getSalary() * 0.02;
    }
}
```

P

P 11

Take a screenshot of one of your projects where you have worked alone and attach the Github link.

		 <p>https://github.com/Chrismlugo/Project-1</p>	
P	P 12	Take screenshots or photos of your planning and the different stages of development to show changes.	

	Unit	Ref.	Evidence	Done
Week 11	I & T		Unit, integration and acceptance testing task B	
	P	P 16	Show an API being used within your program. Take a screenshot of: * The code that uses or implements the API * The API being used by the program whilst running	
	Unit	Ref.	Evidence	Done

Week 13	P	P 1	Take a screenshot of the contributor's page on Github from your group project to show the team you worked with.	
	P	P 2	Take a screenshot of the project brief from your group project.	
	P	P 3	Provide a screenshot of the planning you completed during your group project, e.g. Trello MOSCOW board.	
	P	P 4	Write an acceptance criteria and test plan.	
	P	P 7	Produce two system interaction diagrams (sequence and/or collaboration diagrams).	
	P	P 8	Produce two object diagrams.	
	P	P 9	Select two algorithms you have written (NOT the group project). Take a screenshot of each and write a short statement on why you have chosen to use those algorithms.	
	P	P 17	Produce a bug tracking report	