Module 10 Systems Analysis and Design 2

Module title	Systems Analysis and Design 2
Module NFQ level (only if an NFQ level can be demonstrated)	6
Module number/reference	BSCH-SAD2
Parent programme(s)	Bachelor of Science (Honours) in Computing Science
Stage of parent programme	Stage 2
Semester (semester1/semester2 if applicable)	Semester 1
Module credit units (FET/HET/ECTS)	ECTS
Module credit number of units	5
List the teaching and learning modes	Direct, Blended
Entry requirements (statement of knowledge, skill and competence)	Learners must have achieved programme entry requirements
Pre-requisite module titles	BSCH-SAD1
Co-requisite module titles	None
Is this a capstone module? (Yes or No)	No
Specification of the qualifications (academic, pedagogical and professional/occupational) and experience required of staff (staff includes workplace personnel who are responsible for learners such as apprentices, trainees and learners in clinical placements)	Qualified to as least a Bachelor of Science (Honours) level in Computer Science or equivalent and with a Certificate in Training and Education (30 ECTS at level 9 on the NFQ) or equivalent.
Maximum number of learners per centre (or instance of the module)	60
Duration of the module	One Academic Semester, 12 weeks teaching
Average (over the duration of the module) of the contact hours per week	3
Module-specific physical resources and support required per centre (or instance of the module)	One class room with capacity for 60 learners

Analysis of required learning effort		
	Minimum ratio teacher / learner	Hours
Effort while in contact with staff		
Classroom and demonstrations	1:60	36
Monitoring and small-group teaching		
Other (specify)		
Independent Learning		
Directed e-learning		
Independent Learning		44
Other hours (worksheets and assignments)		45
Work-based learning – learning effort		
Total Effort		125

Allocation of marks (within the module)					
	Continuous assessment	Supervised project	Proctored practical examination	Proctored written examination	Total
Percentage contribution	60%			40%	100%

Module aims and objectives

This module is a continuation module and introduces you to the fundamental concepts of object oriented program design and how to use modelling for constructing complex software systems. As a result, learners develop skills such as communication literacy, critical thinking, analysis, reasoning, and interpretation, which are crucial for gaining employment and developing academic competence. A big emphasis is placed on using UML to module systems and produce designs.

Minimum intended module learning outcomes

On successful completion of this module, the learner will be able to:

1. Apply object oriented systems analysis and design techniques to developing software
2. Work with and use UML for object oriented modelling
3. Develop use cases - both diagrams and narratives
4. Model an overall system using UML class diagrams
5. Model system functionality using UML sequence, collaboration, and activity diagrams
6. Discuss the importance of system operation and ongoing support issues/concerns

Rationale for inclusion of the module in the programme and its contribution to the overall MIPLOs

As we introduce the object orientated paradigm to the learners, they also need to understand how to specify and design systems in this paradigm. This module introduces the concepts of UML and object orientated design to the learners. Appendix 1 of the programme document maps MIPLOs to the modules through which they are delivered.

Information provided to learners about the module

Learners receive a programme handbook to include module descriptor, module learning outcomes (MIMLO), class plan, assignment briefs, assessment strategy and reading materials.

Module content, organisation and structure

Object Orientation

- What is an Object?
- What is a Class?
- What is an Attribute?
- What is a Method?
- What is Encapsulation?

Superclasses and Subclasses

- What is a Message?
- What is Polymorphism?
- What is an Interface?
- What is Component?
- What is a Package?
- Associations and Relationships
- Multiplicities

UML Use Cases

- What is a Use Case?
- What is a Scenario or a Sequence?
- What is an Actor?
- Use Case Deliverables and Artefacts
- Use Case Diagrams
- Definition and Symbols
- Boundary, Relationship, <<include>>, <<extend>>
- Generalization, Abstraction
- Use Case Description/Narrative
- Components of a use case Narrative
- Why use Use Cases?

- System Modelling and the Unified Modelling Language

Systems Analysis - Conceptual Data Modelling using UML

- What is a Class Diagram?
- Elements of a Class Diagram
- Class, Attributes, Operations, Relationships
- Identifier, Multi-Valued and Composite Attributes
- Guidelines for Choosing a Good Identifier
- Relationship Multiplicities
- Associative Classes
- Aggregation and Composition Relationships
- Generalisation Relationships

Systems Analysis - Analysis Classes in UML

- Interaction Diagrams
- Sequence Diagrams
- Entity, Boundary and Control Classes
- What is a Message?
- Elements of a Sequence Diagram
- Collaboration Diagrams
- Activity Diagrams
- Elements of an Activity Diagram
- State Diagrams
- What are Business Rules?
- Type of Business Rules

Systems Design - Architecture

- What is System Architecture?
- The Goal of Good Architecture
- Tiered architecture
- Single-tier, two-tier, three-tier, n-tier
- What is a component?
- Mapping classes from Analysis to Design

Systems Implementation & Operation

- Program Coding
- Code Reuse and Components
- Unit, Integration, System, and UAT Testing review in context
- Installation and Deployment
- System and User Documentation
- Training and Support
- Maintenance
- Measuring and Controlling Maintenance

- Maintenance Cost Factors

Module teaching and learning (including formative assessment) strategy
The module is delivered through a series of lectures. The learners complete a series of worksheets throughout the module that are directly related to the material covered in lectures.

There are two further assignments that test the learners in a larger capacity than the worksheets. Assessment consists of a series of continuous assignments and a final examination.

Timetabling, learner effort and credit
The module is timetabled as one 3-hour lecture per week.

The number of 5 ECTS credits assigned to this module is our assessment of the amount of learner effort required. Continuous assessment spreads the learner effort to focus on small steps before integrating all steps into an e-portfolio to document experience throughout the semester.

There are 36 contact hours made up of 12 lectures delivered over 12 weeks with classes taking place in a classroom. The learner will need 44 hours of independent effort to further develop the skills and knowledge gained through the contact hours. An additional 55 hours are set aside for learners to work on assignments that must be completed for the module.

The team believes that 125 hours of learner effort are required by learners to achieve the MIMLOs and justify the award of 5 ECTS credits at this stage of the programme.

Work-based learning and practice-placement
There is no work based learning or practice placement involved in the module.

E-learning
The college VLE is used to disseminate notes, advice, and online resources to support the learners. The learners are also given access to Lynda.com as a resource for reference.

Module physical resource requirements
Requirements are for a classroom for 60 learners equipped with a projector, and a space to allow the facilitation of group work through movable furniture.

Reading lists and other information resources

Recommended Text

Unhelkar, B. (2018) *Software Engineering with UML*. Boca Raton: Taylor & Francis.

Satzinger, J., Jackson, R. and Burd, S. D. (2015) *Systems Analysis and Design in a Changing World*. Boston: Course Technology.

Secondary Reading:

Seidl, M. (2015) *UML @ Classroom*. New York: Springer.

Podeswa, H. (2008) *The Business Analyst's Handbook*. Boston: Cengage Learning.

Specifications for module staffing requirements

For each instance of the module, one lecturer qualified to at least Bachelor of Science (Honours) in Computer Science or equivalent, and with a Certificate in Training and Education (30 ECTS at level 9 on the NFQ) or equivalent.. Industry experience would be a benefit but is not a requirement.

Learners also benefit from the support of the programme director, programme administrator, learner representative and the Student Union and Counselling Service.

Module Assessment Strategy

The assignments constitute the overall grade achieved, and are based on each individual learner's work. The continuous assessments provide for ongoing feedback to the learner and relates to the module curriculum.

No.	Description	MIMLOs	Weighting
1	A series of worksheets that examine elements of UML	1-6	20%
2	2 take home assignments	1-6	40%
3	Written exam that tests the theoretical aspects of the module	1-6	40%

All repeat work is capped at 40%.

Sample assessment materials

Note: All assignment briefs are subject to change in order to maintain current content.

Worksheets

Activity Diagram Class Exercises

1. Updating of Real Estate Agency Listing

Described below is the process by which a real estate agent updates a listing of property for sale with help of the real estate agency's information system. Create an activity diagram to represent this process.

Real Estate Agent	Listing Management System
1. Enter agent ID	1. Verify agent ID; If incorrect, terminate the session.
2. Enter listing number	2. If the listing is found, display it. Otherwise display: "Error: Enter new #".
3. Enter listing update	3. Verify the agent's privilege to update the listing. If yes, save changes. Otherwise, display: "Error: Not allowed", and end the session.
4. End the session	

2. Parking Booking via Cell Phone

One option customers of the parking space in a metropolitan area have at their disposal is to book parking space via the cell phone. A customer texts his/her registration plate number to a phone number of the line accessing a parking information system. The phone number is specific to a particular parking zone, which can be 1, 2 or 3. Each zone allows for a different parking time and has a different price.

If the customer did not exceed three allowed bookings in a row at that location and time, the parking system confirms via a return text message that the parking spot has been booked at a certain time. Ten minutes before the parking expiry, the parking system warns the customer via a text message, asking if an extension is desired. The customer can extend the parking simply by hitting the reply key on their phone, or decline to extend the parking. Create an activity diagram to represent this process.

3. Withdrawal Process with Automatic Teller Machine

When a customer wants to withdraw the money from an Automatic Teller Machine (ATM), the customer inserts an ATM card (debit card) or a bank credit card in the ATM. The ATM validates the card expiration date and, if it is OK, prompts for a PIN entry. In case of an invalid card, the card is returned back and the procedure ends.

After the customer's entering of a PIN, the ATM validates it. This involves a run of a PIN validation step by an account management system at the customer's bank. In case of an invalid PIN, the card is returned back and the procedure is canceled. Provided the PIN is valid, the ATM prompts the customer to enter the amount to be withdrawn. The customer enters the desired amount. The balance checking involves again the customer's bank.

If the balance is sufficient, the ATM disburses cash, returns the card, and prints the receipt. If the balance is insufficient, the customer may decide to continue by entering the amount again or not to continue. In the latter case, the ATM returns the card, and prints the receipt.

4. Job Search Process (open-ended)

Create an activity diagram for a job search process. You can make any assumptions you deem appropriate. Your diagram can include job search in different advertising venues, resume and application creation and submission, interviewing… all the way to a hiring decision.

Keep it simple, and start with a list of key activities. Who performs those activities? Then, expand analysis asking if there are any decisions, parallel activities, loops, and any other component?

Once you have components identified, focus on their order. Note that a challenge may be to keep at the same level of analysis without frilling down into some steps that may be sub-processes on their own.

5. Analyze functionality of some social medium you are familiar with and depict one function via an activity diagram. For example, there may be a login step, performed by the user and supported by the social medium system; then, there may be steps for displaying the content, reading it, uploading new content, etc.

Class Diagram Class Exercises

The I.T. department of a medium-sized consulting firm requires an inventory system. This system would allow them to individually track every piece of computer equipment used by the company. This includes computers, monitors, printers, etc. This is especially important given that the company has several offices located across North America.

The system should keep track of the following characteristics of equipment:

- Its type (such as Computer, Monitor, Printer)
- Its serial number (such as N3JEKW357EA)
- Its model numbers
- Information about its purchase (such as date, store, warranty expiration)
- Who is currently assigned to this piece of equipment
- Its location (such as Montreal Office, Toronto Office)

In addition,

- every piece of computer equipment that needs to be tracked has a special bar code sticker on it which uniquely identifies it in the system. This bar code is a random unique number assigned by the system.
- a piece of equipment can only be assigned to one employee at a time.

The system also keeps track of employees. Although not as complete as a Human Resources (HR) system, it does record the following information :

- Name of the employee
- Employee gender (male/female)
- His/her title
- His/her department
- His/her supervisor
- His/her location (such as Montreal Office, Toronto Office)
- His/her coordinates (such as address, phone, cell)
- List of pieces of equipment currently assigned to him/her
- List of pieces of equipment assigned to him/her in the past (and when!)

Some information about locations is also stored:

- Name of the location
- Number of offices in that location
- List of (references to) employees in that office
- Employee responsible for I.T. maintenance at that office and his/her coordinates
- List of equipment found at that location

Task 1

Draw a UML class diagram to model the above inventory system.

- Possibly start by informally drawing the diagram on paper. Do not need hand it in.
- Draw your diagram using a UML design tool of your choice.
- Submit your design drawn in your chosen tool
- You will be assessed on the cardinalities, attributes, proper syntax, correct association types, ...

Task 2

- From the class diagram designed in UML
- Draw an object diagram which shows the objects and their links at run-time. You can limit yourself to 20 objects.

UML SOFTWARE

Here is a non-exhaustive list of some UML tools you could use to do this assignment.

- BoUML
- MagicDraw
- Dia
- ArgoUML
- UMLet
- Violet
- TOPCASED (plugin for Eclipse. beware: install all pre-requisite plugins first)
- Visio
- Poseidon UML

You can find more UML tools online

Use Case Worksheets

USE CASE 1	Register Student	
Goal in Context	To register a student for a water safety class	
Scope & Level	Registration subsystem Primary User Task	
Preconditions	Student previously registered in system.	
Success End Condition	Student registered and details stored	
Failed End Condition	Student not registered and details not stored	
Primary, Secondary Actors	Voluntary Registration assistant	
Trigger	Register selected	
DESCRIPTION	**Step**	**Action**
	1	Student details [number/ name] entered
	2	Full student details displayed [address, phone no, email, qualifications, dates]
	3	List of available courses/levels displayed
	4	Course/level selected
	5	Payment noted
	6	Student registered and put in class.
	7	Class time displayed
EXTENSIONS	**Step**	**Branching Action**
	1a	Student not registered - <u>create student (UC2)</u>
	4a	Course/level not available or full up
	7a	Class not timetabled yet—inform student
SUB-VARIATIONS		**Branching Action**
	1-5	Quit without saving
OPEN ISSUES		Do schedule and classes need to be created before registering student? What are the upper limits to numbers and who specifies them?

USE CASE 2	Create Student	
Goal in Context	To create a student record	
Scope & Level	Registration subsystem Primary User Task	
Preconditions		
Success End Condition	Student details stored	
Failed End Condition	Student details not stored	
Primary, Secondary Actors	Voluntary Registration assistant	
Trigger	Create selected	
DESCRIPTION	**Step**	**Action**
	1	Student form displayed
	2	Full student details entered [name address, phone no, mobile number, email, parents name and phone no., if a minor]
	3	Student number generated and displayed, record stored
	4	Any existing qualifications are added [qualifications, dates]
EXTENSIONS	**Step**	**Branching Action**
SUB-VARIATIONS		**Branching Action**
	1-4	Quit without saving
OPEN ISSUES		Do we need to store any more details on existing qualifications e.g. examiner?

USE CASE 3	Register Trainer for session	
Goal in Context	To register a trainer to teach in a session	
Scope & Level	Registration subsystem Primary User Task	
Preconditions	Trainer previously registered in system as trainer or student.	
Success End Condition	Trainer registered and details stored	
Failed End Condition	Trainer not registered and details not stored	
Primary, Secondary Actors	Water safety manager	
Trigger	Register selected	
DESCRIPTION	**Step**	**Action**
	1	Trainer details [number/ name] entered
	2	Full trainer details displayed [address, phone no, email, qualifications, dates]
	3	Training record to date displayed [session and course taught, supervisor- if applicable]
	4	Garda vetting details displayed and updated if necessary
	5	
	6	
	7	
EXTENSIONS	**Step**	**Branching Action**
	1a	Trainer not registered - <u>create trainer</u>
	1b	Trainer was student – retrieve details and alter status
	4a	Course/level not available or full up
	7a	Class not timetabled yet—inform student
SUB-VARIATIONS		**Branching Action**
	1-5	Quit without saving
OPEN ISSUES		Do schedule and classes need to be created before registering student? What are the upper limits to numbers and who specifies them?

USE CASE 4	Create Trainer	
Goal in Context	To create a trainer record	
Scope & Level	Registration subsystem Primary User Task	
Preconditions		
Success End Condition	Trainer details stored	
Failed End Condition	Trainer details not stored	
Primary, Secondary Actors	Voluntary Registration assistant	
Trigger	Create selected	
DESCRIPTION	**Step**	**Action**
	1	Trainer form displayed
	2	Full trainer details entered [name address, phone no, mobile number, email, parents name and phone no., if a minor]
	3	Trainer number generated and displayed, record stored
	4	Qualifications are added [qualifications, dates, body]
	5	Garda vetting details added
EXTENSIONS	**Step**	**Branching Action**
SUB-VARIATIONS		**Branching Action**
	1-4	Quit without saving
OPEN ISSUES		Do we need to store any more details on existing qualifications e.g. examiner, or equivalence if with a different body?

USE CASE #5	Create new schedule	
Goal in Context	To create a schedule for water safety classes	
Scope & Level	Scheduling subsystem Primary User Task	
Preconditions	None	
Success End Condition	Schedule created and details stored	
Failed End Condition	Schedule not created	
Primary, Secondary Actors	Water Safety Manager	
Trigger	Create schedule selected	
DESCRIPTION	**Step**	**Action**
	1	Schedule details [name (e.g. Spring 2011) , start and end dates, day and times] are specified. are specified.
	2	Blank schedule created and displayed
	3	Classes are added to schedule
	4	Class details [start and end time and date] are specified.
	5	Qualification and level selected from list
	6	<u>Assign trainer to class (UC5)</u>
	7	Details confirmed and Class added.
	8	Schedule displayed for confirmation
	9	Schedule saved
	Step	**Branching Action**
EXTENSIONS	1a	Schedule already created with that name or for those dates. Open that schedule
	4a	Class details invalid
	6a	Trainer not available – leave unassigned
	9a	Print option selected -<u>print schedule</u>
		Branching Action
SUB-VARIATIONS	1b-5b	Quit without saving
OPEN ISSUES		

USE CASE 6	Edit Schedule	
Goal in Context	To edit a water safety schedule	
Scope & Level	Scheduling subsystem Primary User Task	
Preconditions	Schedule exists	
Success End Condition	Schedule amended and details stored	
Failed End Condition	Schedule not amended	
Primary, Secondary Actors	Water Safety Manager	
Trigger	Edit Schedule selected	
DESCRIPTION	**Step**	**Action**
	1	Schedule Selected
	2	Class created or selected
	3	Class details [start and end time and date] are specified.
	4	Qualification and level selected from list
	5	Trainer selected from available trainer list
	6	Class amended
	7	Schedule amended
EXTENSIONS	**Step**	**Branching Action**
	1a	Schedule doesn't exist -Create new schedule [UC3]
SUB-VARIATIONS		**Branching Action**
	1-5	Quit without saving
OPEN ISSUES		

USE CASE 7	Assign trainer to class	
Goal in Context	Assign a trainer to a class on a water safety schedule	
Scope & Level	Scheduling subsystem Primary User Task	
Preconditions	Schedule exists, trainer exists and is qualified	
Success End Condition	Schedule amended and details stored	
Failed End Condition	Schedule not amended	
Primary, Secondary Actors	Water Safety Manager	
Trigger	Assign trainer selected	
DESCRIPTION	**Step**	**Action**
	1	Schedule selected from list
	2	Class selected from list
	3	Trainer selected from list
	4	System checks if trainer eligible for this class
	5	Details displayed for confirmation
	6	Trainer added to class
EXTENSIONS	**Step**	**Branching Action**
	2a	Class not created or qualification not assigned- Class details [start and end time and date] are specified. and Qualification and level selected from list
	3a	Trainer not in system yet – register trainer
	4a	Trainer not eligible – select another class, or another trainer
SUB-VARIATIONS		**Branching Action**
	1-5	Quit without saving
OPEN ISSUES		

USE CASE 8	Log exam results	
Goal in Context	Enter results of examinations	
Scope & Level	Examination subsystem Primary User Task	
Preconditions		
Success End Condition	Exam results stored for a course	
Failed End Condition		
Primary, Secondary Actors	Water safety manager or authorized person	
Trigger	Exam results	
DESCRIPTION	**Step**	**Action**
	1	Schedule displayed
	2	Class selected
	3	Examination details – [examiner, date] entered
	4	Students displayed with result slots
	5	Results entered
	6	Results for each student are stored
	7	Trainers details displayed and confirmed
	8	Trainer's trainings record updated
	9	Results printed to exam form.
EXTENSIONS	**Step**	**Branching Action**
	5a	Student did not do exam, result not entered, nothing stored
	7a	Trainer details incorrect – record not updated and correction made [assign trainer to class]
	9a	Printing not required – results emailed
SUB-VARIATIONS		**Branching Action**
	1-4	Quit without saving
OPEN ISSUES		Can the results be transferred electronically?

Take Home Assignments

System Analysis and Design 2 - Assignment 1

The requirements model should have delineated the **scope** of the system you hope to develop, and hopefully has ironed out many ambiguities. This means that it will clearly show, in a way that most people can understand, what exactly the system you hope to develop will be required to do and how each function will work from the user's perspective. In this assignment, you are required to produce a class diagram and simple collaboration (robustness) diagrams for the water safety application. I will put use case templates on Moodle.

Tasks

1. Make sure you clearly understand what must be done in the water safety system.
2. Identify a set of candidate **entity classes**. List their **attributes** and specify **relationships** between these classes, including multiplicity (e.g. 1 to 0..*) . If there are many-many relationships add a linking class to resolve these. It might help to think of this linking class as a contract or transaction– the thing that relates the other classes.
3. Draw a rough conceptual class diagram and for each class note down any significant operations. Remember that all classes will have a constructor to create objects of that class, a destructor to dispose of them and a number of get (query) and set operations to change the values of attributes. These common operations are ***not shown here*** to save space.
4. Work on sections of your diagram to clarify issues and ensure that it is complete.
5. Identify any similarities between classes and note where you might employ inheritance.
6. Sketch out a rough class diagram showing where inheritance might be employed.
7. Draw **collaboration or robustness diagrams** for each use case to show which classes could communicate to implement each use case. Identify control and boundary(interface) classes and specify their attributes and operations.
8. Fill in class responsibility cards and "walk through" each use case to ensure that the functionality has been achieved.

To be submitted:

1. Class diagram showing entity classes and relationships – you can show inheritance relationships on this diagram if you wish, but if it gets too messy you can show them on a separate diagram. Attributes and operations should also be shown. However, if these make the diagram too large or messy, again you can put them in separate diagrams.
2. Collaboration/robustness diagrams for five significant use cases
3. Class diagrams of other interface and control classes.
4. Any **rough drafts** you might have: CRC cards, rough class diagrams you use to get your answer.

Assignment 2: Detailed OO design

Aims

1. To complete the detail of the analysis and to detail the design of your software.
2. To design a user interface for registration and scheduling sub-systems.
3. To implement part of the system, following principles of software design.

Example Use Cases

1. Create student / Register student for a class
2. Create schedule of classes
3. Register trainer/assign trainer to a class
4. Enter exam results.

To be submitted:

5. Detailed OO design. Finalised class diagrams showing **types** of attributes and **operation signatures** showing parameter types. Ensure that you have dealt with multiplicity by including collection classes where necessary.

6. A **sequence diagram** for each use case you intend to implement.

7. **User interface design** e.g. using storyboards, sketches, task analysis, state machine diagrams, prototypes. Your designs should demonstrate consideration of key principles of User Interface design.

8. Implementation of entity, control and UI classes in java showing adherence to **design principles** – **loose coupling**, **strong cohesion** and demonstrating the use of **clean code**. You can use dummy data for the data access layer.

GRIFFITH COLLEGE DUBLIN

QUALITY AND QUALIFICATIONS IRELAND

EXAMINATION

SYSTEMS ANALYSIS & DESIGN 2

Lecturer:

External Examiner:

Date: XX/XX/XX **Time: XX.XX**

THIS PAPER CONSISTS OF FOUR QUESTIONS
SECTION A – COMPULSORY
SECTION B – 2 OUT OF 3 QUESTIONS TO BE ATTEMPTED

SECTION A – Compulsory

QUESTION 1

(a) Describe the term '*extending a use case*'. Illustrate this in UML, using a relevant example.

(5 marks)

(b) What is the purpose and advantage of creating a sequence diagram in UML?

(5 marks)

(c) What is multiple-inheritance? Is it possible in Java?

(5 marks)

(d) Explain the term *method overloading* and how it is used in constructors. Give an example in Java.

(5 marks)

(e) Distinguish the difference between private, protected and public access modifiers in Java.

(2 marks)

(f) Define an abstract class.

(5 marks)

(g) What is meant by java 'bytecodes'? How does this translation method differ from other programming languages?

(5 marks)

(h) Define an interface?

(5 marks)

(i) What is a 'wrapper' method? Give an example?

(5 marks)

(j) Briefly outline ways in which object orientated design is used to improve code reusability?

(5 marks)

Total (50 marks)

SECTION B – 2 QUESTIONS TO BE ATTEMPTED

QUESTION 2

(a) A library system has engaged you to make recommendations to automate the process of borrowing a library book. The system will use the Unified Modelling Language (UML).

Construct the following documents for the proposed system:

(i) Use Case Diagram

(5 marks)

(ii) Class Diagram

(5 marks)

(iii) Sequence Diagram

(5 marks)

(iv) Activity Diagram

(5 marks)

(b) Explain the benefits, or otherwise, of an object oriented approach utilising UML in this systems development.

(5 marks)

Total (25 marks)

QUESTION 3

a) Define a System Architecture? **(3 marks)**

b) Outline and describe 5 components of a system architecture **(5 marks)**

c) Outline the goals of a Good Architecture **(5 marks)**

d) Distinguish between 1-Tier, 2-Tier and 3 Tier Architectures **(12 marks)**

Total (25 marks)

QUESTION 4

(a) Explain **four** of the fundamental concepts of object oriented design.

(10 marks)

(b) Briefly outline the need for **object oriented design**

(5 marks)

(c) Outline the **role** of the Unified Modelling Language in Object Oriented Design.

(5 marks)

(d) Explain what is meant by a **class** in Object Oriented Design.

(5 marks)

Total (25 marks)