

Systems Design and Databases (CIS1018-N)

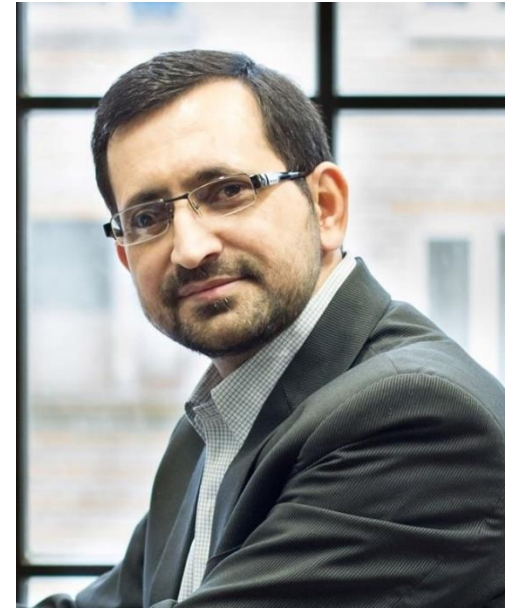
Week 1

Module Introduction,

Systems Design, Introduction to Databases - DDL, DML, DCL & TCL

Requirement List & MoSCoW & User Story

Module Leader & Lecturer: Dr Yar Muhammad
Email: Yar.Muhammad@tees.ac.uk
Office: G0.39 (Greig Building)



Tutor:

- Dr Mengda He
 - Mr Mansha Nawaz
 - Mr Vishalkumar Thakor
-
- See Blackboard Ultra for online materials: <https://bb.tees.ac.uk/>

Lectures & IT Labs

Lectures – Dr Yar Muhammad	Tuesdays @ 2-3 pm	Thursdays @ 1-2 pm
Week 1 – Week 12	CL1.87	OL7

Tutor – Thursday	IT Lab Session Room #: OL8
Mr Mansha Nawaz M.Nawaz@tees.ac.uk	Time: 9 – 10 am Room #: OL8
Mr Mansha Nawaz M.Nawaz@tees.ac.uk	Time: 3 – 4 pm Room #: IT2.42

Tutor – Friday	IT Lab Session Room #: OL3
Dr Yar Muhammad Yar.Muhammad@tees.ac.uk	Time: 9 – 11 am & 11 am – 1 pm
Dr Mengda He M.He@tees.ac.uk	Time: 9 – 11 am
Mr Vishalkumar Thakor V.Thakor@tees.ac.uk	Time: 11 am – 1 pm & 1 – 3 pm
Mr Mansha Nawaz M.Nawaz@tees.ac.uk	Time: 1 – 3 pm

This Week's Agenda

Module Introduction

- Get familiar with the module aims and learning outcomes.
- Familiarise ourselves with how this module will be delivered.
- Take a look at how the module will be assessed.

Jumping to Systems Design & Databases

- Systems Design,
- Introduction to Databases
- DDL, DML, DCL & TCL
- Requirement List & MoSCoW
- Wireframe Design & Template
- User Story

Module Aims

- Introduce and explore the **systems development lifecycle and methods**.
- Introduce the students to the **Unified Modelling Language (UML)** and other diagramming techniques.
- Develop database design and implementation skills.
- Introduce the student to a tools-driven approach to database development.



Personal and Transferable Skills

- PTS1. Understand the use of appropriate process models to specify, design, implement and evaluate an Information System.
- PTS2. Articulate a design derived from requirements gathering within an agreed timescale.
- PTS3. Participate in team activities to achieve a common objective and reflect on your own and the team's performance as future computing professionals.

Research, Knowledge and Cognitive Skills

- RKC1. Identify one or more potential solutions to a project brief through the development of system models.
- RKC2. Select and use appropriate methods and tools to design and develop a solution that satisfies user requirements.

Professional Skills

- PS1. Demonstrate industry standard communication skills in key activities such as interviewing a stakeholder, documenting the proposed new system and reviewing the design.
- PS2. Design and construct a relational database as a solution to a given Information System problem.
- PS3. Demonstrate an understanding of appropriate data protection legislation/regulations.

- **Element # 1 - UML Portfolio -> 50% (Teamwork)**

- There are a number of Systems Design (SD) Challenge Deliverables you will need to submitted in the form of a UML Portfolio (for more detail, see *ICA Specification document, NHS Family Voice - UML Case Study and Systems Design Report*).
- Each member of the Teams needs to be involved in developing a UML Solution and these activities have been scheduled in your Labs.

- **Element # 2 - Database Portfolio -> 50% (Individual work)**

- Based on the any of the ICA designated SQL Server Database available on Blackboard.
- Note:
 - You must not use the teaching sample database TSQL.
 - The component is individual Work

More detail to follow ICA brief documents

Lectures

- Introduce you to theoretical concepts through practical demonstrations in the lectures

Tutorial Activities


- Solve problems using the concepts introduced in the lectures.
- As you progress, the problems will increase in complexity.
- Provide an opportunity to work on the assessment later in the semester.
- Receive regular formative feedback from your tutors and each other to reflect upon and take forward to future activities.

Schedule 1/3

* Subject to change

Systems Design and Databases CIS1018-N Weekly Plan for the Activities

Systems Design - UML

Week	Lecturer	Lecture Demo	Lab Exercises & Solutions	ICA Tasks:
01 	<ul style="list-style-type: none">Module Introduction,System Design,Introduction Databases (DDL, DML, DCL, TCL)	<ul style="list-style-type: none">Requirement List &<u>MoSCoW</u> Wireframe Design & Templates,User Stories	<ul style="list-style-type: none">Team Setup,Hands-on to collect/pick the Requirements from <u>MoSCoW</u> and write Writing User stories on eachTutorial 1	Requirements List & <u>MosCOW</u> , User stories
02	<ul style="list-style-type: none">UML and UML Tool,	<ul style="list-style-type: none">Use Case Diagrams from Requirements List and Wireframe	<ul style="list-style-type: none">Hands-on Use Case Diagrams ActivitiesTutorial 2	Each Wireframe has associated Use Case Activity Deadline for Team Setup is Week # 2, by Friday 07/10/2022 before 4pm
03	<ul style="list-style-type: none">Sequence Diagrams	<ul style="list-style-type: none">Class Diagrams	<ul style="list-style-type: none">Hands-on Sequence & Class Diagrams ActivitiesTutorial 3	Each Wireframe has associated Sequence and Class Diagrams
04	<ul style="list-style-type: none">Entity Relationship Diagrams (ERD) A Data Modelling Case Tool for Relational Databases	<ul style="list-style-type: none">Introduction to SQL ServerWalk-through: SQL Quick Guide 1 - How to use SSMS to build Databases	<ul style="list-style-type: none">Tutorial 4Lab Resources: SQL Quick Guide 1	Each Wireframe has associated Class Diagram

Schedule 2/3

Databases - TSQL

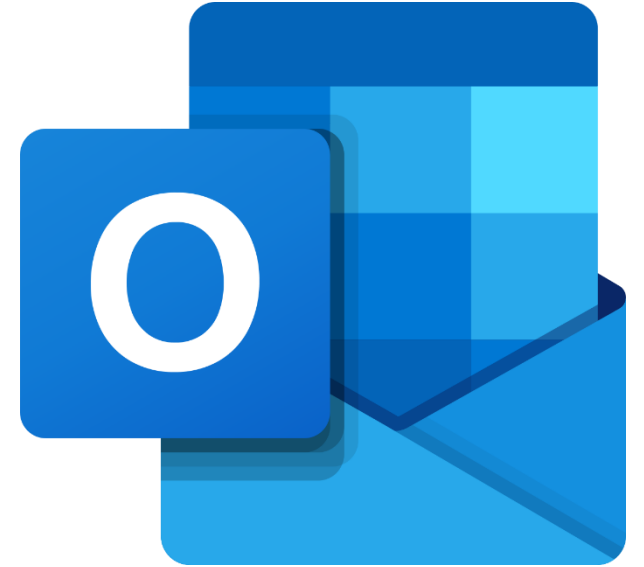
Week	Lecturer	Lecture Demo	Lab Exercises & Solutions	ICA Tasks:
05	<ul style="list-style-type: none"> Querying with Select 	Demo A – Writing Simple SELECT Statements Demo B/C – Eliminating Duplicates with DISTINCT Demo D - Writing Simple CASE	<ul style="list-style-type: none"> TSQL-Mod03 Lab-Exercise 1-4 Tutorial 5 	SQL Task A: TSQL03 Querying with Select <ul style="list-style-type: none"> Writing Simple SELECT Statements Eliminating Duplicates with DISTINCT Using Column and Table Aliases Writing Simple CASE Expressions
06	<ul style="list-style-type: none"> Querying with Multiple Tables 	Demo B – Relating 2 or more tables – Joins & Joining multiple tables – inner, <u>outer</u> and cross.	<ul style="list-style-type: none"> TSQL-Mod04 Exercise 1-5 Tutorial 6 	SQL Task B: TSQL04 – Querying with Multiple Tables <ul style="list-style-type: none"> Relating 2 or more tables – Joins Joining multiple tables – inner, <u>outer</u> and cross.
07	<ul style="list-style-type: none"> Sorting and Filtering Data 	Demo A – Sort with ORDER BY Demo B – Filter with WHERE Clause Demo C – Filtering with Top OffsetFetch Demo D – Handling NULL	<ul style="list-style-type: none"> TSQL-Mod05 Exercise 1 – 4 Tutorial 7 	SQL Task C: TSQL05 – Sort and Filtering Data <ul style="list-style-type: none"> Sort with Order By Filter with <u>Where By</u> Filter with top <u>offsetfetch</u> Handling Nulls
Submission ICA 1 (Group Submission) -> Deadline is Wednesday 16/11/2022 before 4pm				
08	<ul style="list-style-type: none"> Working with SQL Server Data 	Demo A - Conversion in a Query Demo B - collation in a query Demo C - date and time functions	<ul style="list-style-type: none"> TSQL-Mod06 Exercise 1 – 4 Tutorial 8 	SQL Task D: TSQL06 – Working with SQL Server Data <ul style="list-style-type: none"> Conversion in a Query collation in a query date and time functions

09	<ul style="list-style-type: none"> Using DML to modify Data 	Demo A - Adding Data to Tables Demo B - Modifying and Removing Data Demo C - Generating Automatic Column Values	<ul style="list-style-type: none"> TSQL-Mod07 Exercise 1 – 2 Tutorial 9 	SQL Task E: TSQL07– Using DML to Modify Data <ul style="list-style-type: none"> Adding Data to Tables Modifying and Removing Data Generating Automatic Column Values
10	<ul style="list-style-type: none"> Using built in Functions 	Demo A – Scalar Functions Demo B – Cast Functions Demo C – If Functions Demo D – <u>IsNull</u> Functions	<ul style="list-style-type: none"> TSQL-Mod08 Exercise 1 – 3 Tutorial 10 	SQL Task F: TSQL08– Using Built-In Functions <ul style="list-style-type: none"> Writing Queries with Built-In Functions Using Conversion Functions Using Logical Functions Using Functions to Work with NULL
11	<ul style="list-style-type: none"> Walk through SQL Quick Guide 2 - Create a Tables and Relationships via SSMS GUI 	<ul style="list-style-type: none"> Walk through: SQL Quick Guide 3 - Create Query, View through Designer 	Hands-on: <ul style="list-style-type: none"> SQL Server Quick Guide 2 	SQL Server – Introduction to SQL Server and SSMS
12	Support	Support	Hands-on: <ul style="list-style-type: none"> SQL Server Quick Guide 3 	SQL Server – Introduction to SQL Server and SSMS
Submission ICA 2 (Individual Submission) -> Deadline is Wednesday 11/01/2023 before 4pm				

Getting Support

I'm available on email

- You get in touch via email at:
Yar.Muhammad@tees.ac.uk
- I'll aim to get back to you as soon as I possibly can.



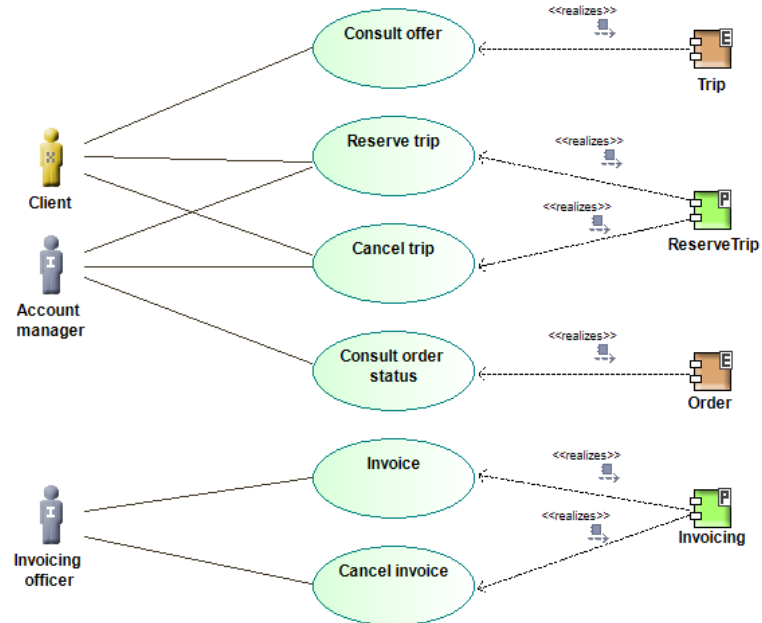
Recommended Reading

- [Learning UML 2.0](#), Miles, Russ; Hamilton, Kim, 2006
- [UML 2.0 in a nutshell](#), Pilone, Dan; Pitman, Neil, 2005
- [System analysis and design UML version 2.0: an object-oriented approach](#), Tegarden, David Paul; Tegarden, David; Dennis, Alan; Wixom, Barbara Haley 2012 - 4th ed., International student version
- [A first course in database systems](#), Ullman, Jeffrey D; Widom, Jennifer, 2014 - 3rd ed.
- [Welcome to the SQL Server Community](#)
- [Microsoft SQL documentation - SQL Server | Microsoft Docs](#)

System Design and Databases

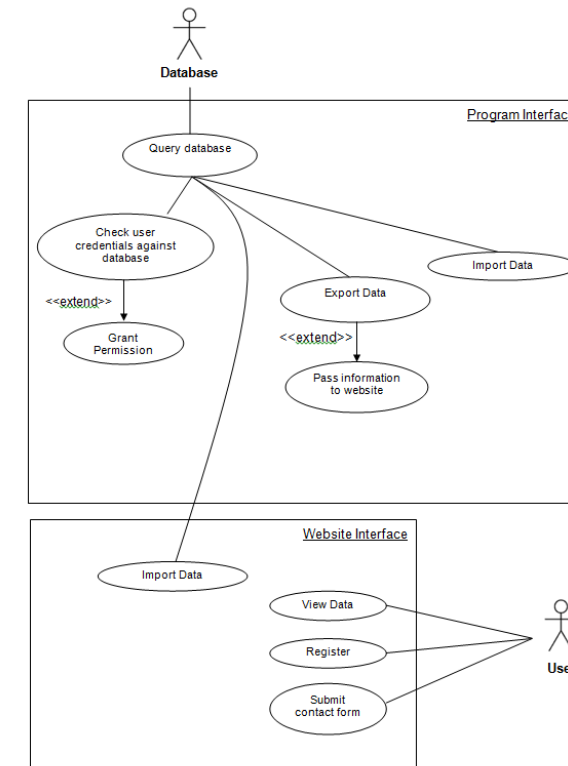
Systems Design

- Systems design is the process of defining elements of a system like modules, architecture, components and their interfaces and data for a system based on the specified requirements.



Databases

- A database is an organized collection of structured information, or data, typically stored electronically in a computer system.



What is Databases - Definition 1

- A database is a shared collection of logically related data, that store to meet the requirements of different users an organization
- Data :- Raw facts

Database Definition 2

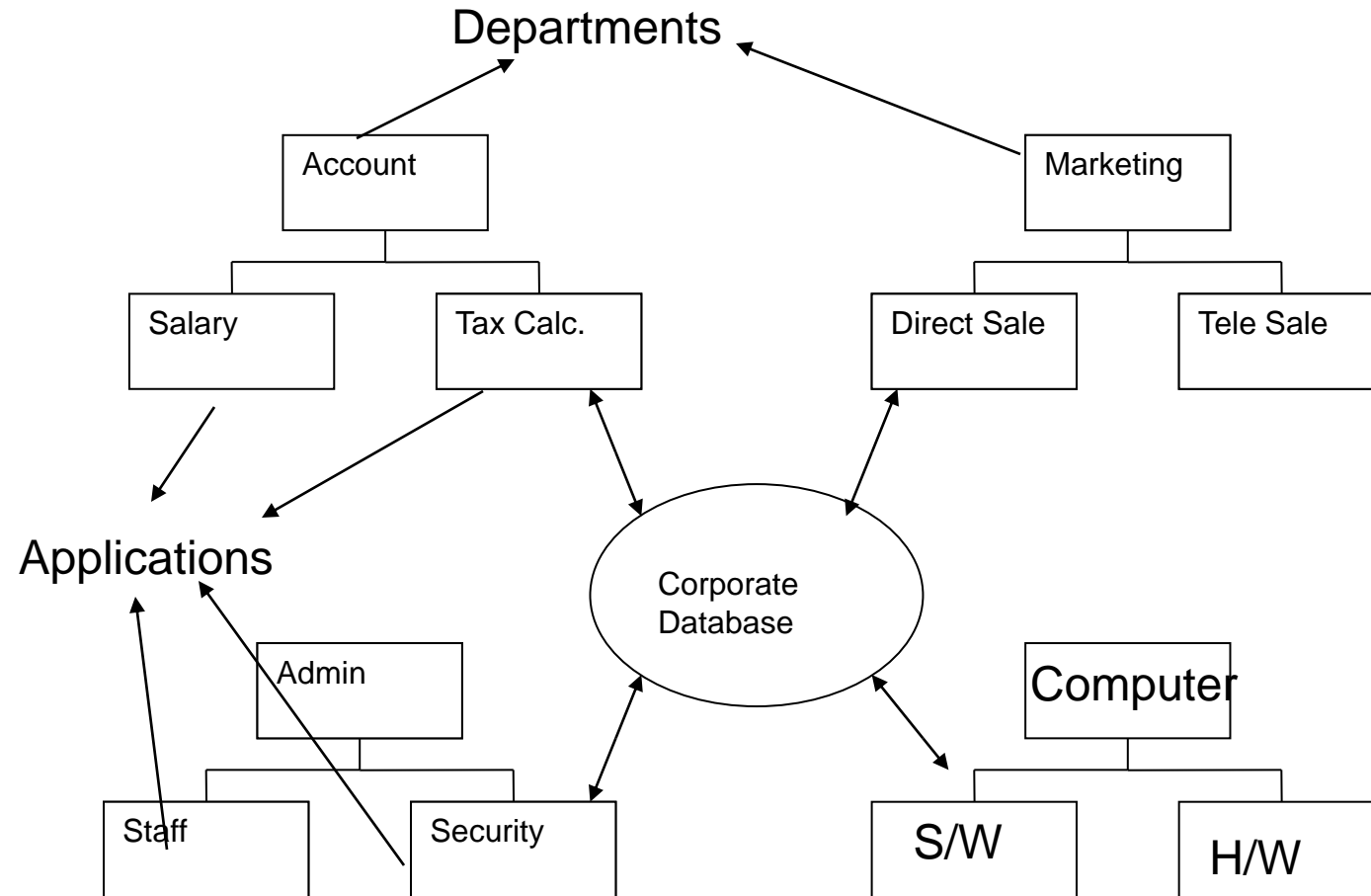
- A database is a self-describing collection of integrated records.
- Database structure, schema, metadata
- integrated record link each other

Database Definition 3

- A database models a particular real world system in the computer in the form of data
- Real World System like Reservation System, ATM System, e-vision so on which work in real environment because different people and operations are involved.

The Concept of a Shared Organizational Database

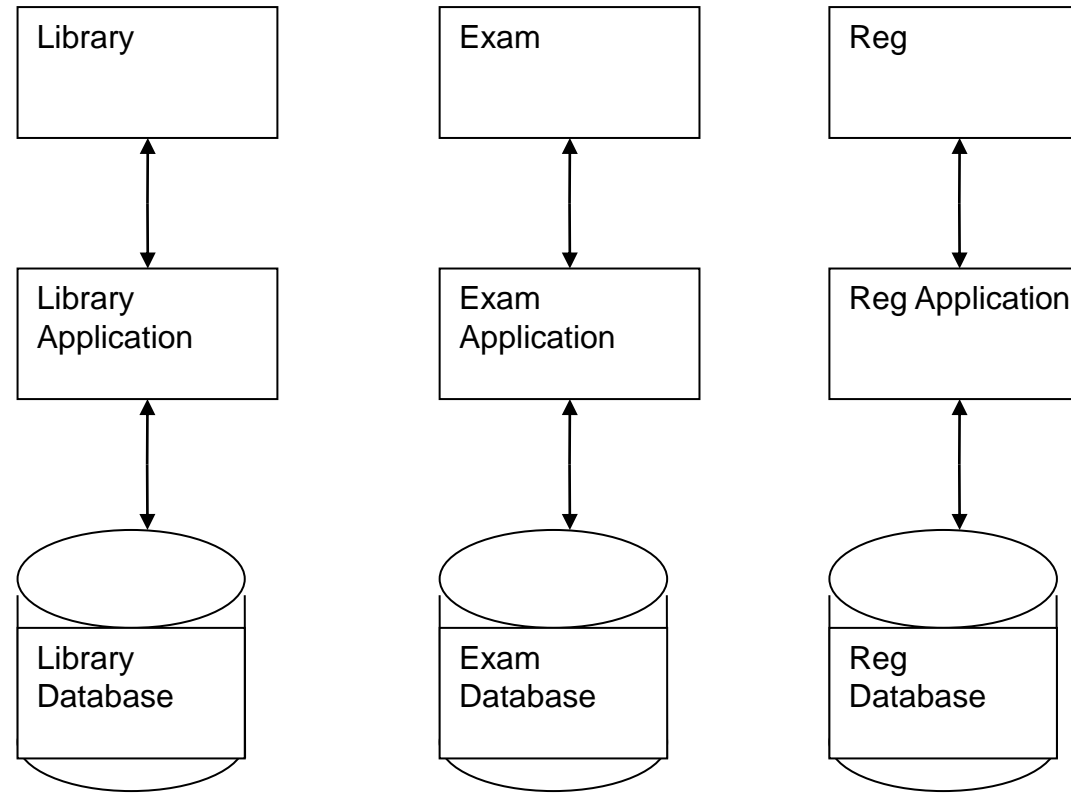
- Each Dept. is separated but database is single, this is concept of shared org database



File Processing System/ Environment

- FPS is a collection that perform service for end user for producing reports

University File Processing System/ Environment



Problems?

- Individual data is maintained
- Data Interdependent (make changes in all)

Problems

- **Duplication of Data**
- **Inconsistency**
- **Storage**

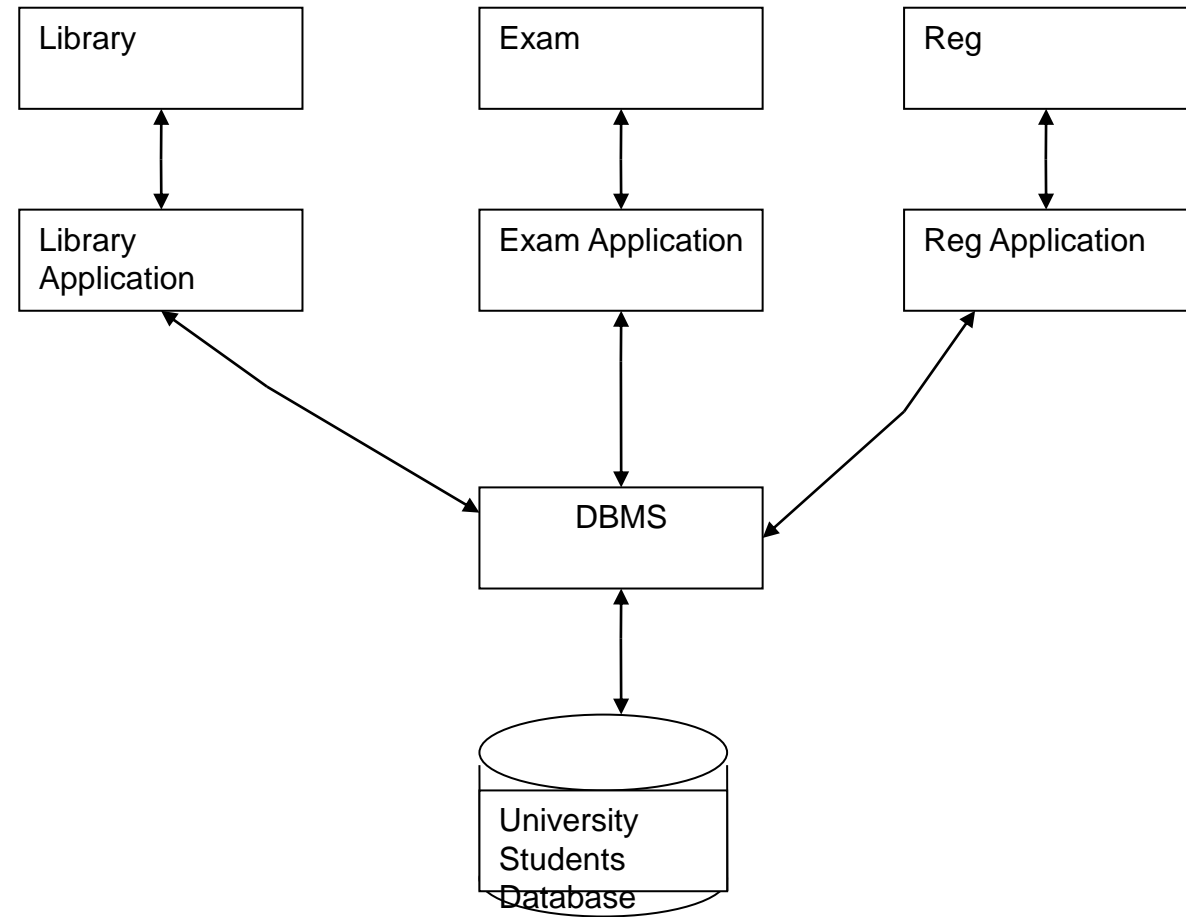
<u>Library</u>
<i>Reg_No</i>
<i>Name</i>
<i>Father Name</i>
<i>Books Issue</i>
<i>Fine</i>

<u>Exam</u>
<i>Reg_No</i>
<i>Name</i>
<i>Address</i>
<i>Class</i>
<i>Semester</i>
<i>GPA</i>

<u>Reg</u>
<i>Reg_No</i>
<i>Name</i>
<i>Father Name</i>
<i>Phone</i>
<i>Address</i>
<i>Class</i>

Advantage of Database Approach

- **Data Sharing**
- **Control Redundancy**
- **Data Independence**
- **Better Data Integrity.**



Further Advantages of Database System

- Data Consistency
- Better data security
- Faster development of new applications
- Economy scale
- Better concurrency control
- Better backup and recovery procedures

Data Consistency

- **All the occurrence have same value** for any specific data item
- Data inconsistency leads to a number of problems
 - Loss of information
 - Incorrect result.

Better Data Security

- All application programs **access data through DBMS**
- DBMS very efficiently check which user is performing which action and accessing which part of data.

Fast Application Development

- Due to following reason
 - The **data needed for the new application already resides in the database.**
 - The data might not already reside in the database, but it could **be derived from the data present in the database**

Better Concurrency Control

- Access a database from as number of point simultaneously.
- Database access completed correctly and transparently
 - ATM Machine

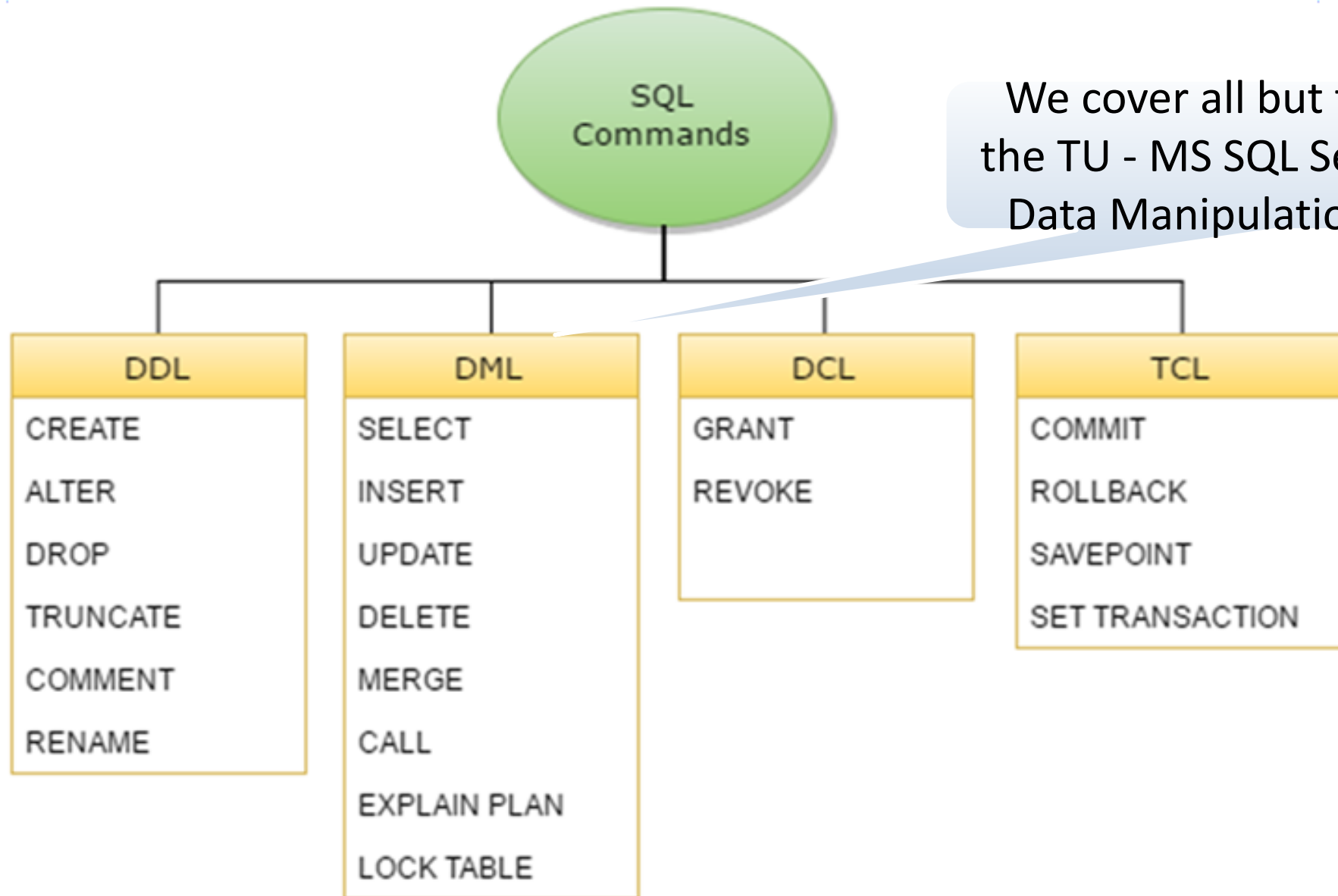
Better Backup and Recovery

- Data is very important resource for organization.
- It is necessary to take **periodic backup of data**
- Sudden due **to disastrous situation** the **database crashes (improper shutdown, invalid disk access, etc)**, should be able to recover the database

Disadvantages

- Higher costs
- Conversion cost
- More difficult recovery

TSQL Statement Types



SQL Statement Types – Transactional SQL or TSQL

Data Manipulation Language (DML)	Data Definition Language (DDL)	Data Control Language (DCL)
Statements for querying and modifying data: <ul style="list-style-type: none">• SELECT• INSERT• UPDATE• DELETE	Statements for defining database objects: <ul style="list-style-type: none">• CREATE• ALTER• DROP	Statements for assigning security permissions: <ul style="list-style-type: none">• GRANT• REVOKE• DENY



Primary focus on DML

TSQL Statement Types: Data Definition Language DDL

- **DDL** is Data Definition Language : it is used to define data **structures**.
- **Data Definition Language DDL**: is the TSQL script code which deals with database schemas and descriptions, of how the data should reside in the database.
- **DDL is the instruction set to create and maintain Relational DB and its Tables.**

CREATE – to create database and its objects like (table, index, views, store procedure, function and triggers).

ALTER – alters the structure of the existing database.

DROP – delete objects from the database.

TRUNCATE – remove all records from a table; also, all spaces allocated for the records are removed.

COMMENT – add comments to the data dictionary.

RENAME – rename an object.

TSQL Statement Types: Data Manipulation Language DML

- **DML** is Data Manipulation Language : it is used to manipulate **data itself**.
- **Data Manipulation Language DML** which deals with data manipulation, and includes most common SQL statements such SELECT, INSERT, UPDATE, DELETE etc, and it is used to store, modify, retrieve, delete and update data in database.
- **DML commands help query and maintain the Relational DB Tables and its data. It helps relate table data via primary and foreign key mapping.**

SELECT – retrieve data from one or more tables.

INSERT – insert data into a table.

UPDATE – updates existing data within a table.

DELETE – delete all records from a table.

MERGE – UPSERT operation (insert or update)

CALL – call a PL/SQL or Java subprogram.

EXPLAIN PLAN – interpretation of the data access path.

LOCK TABLE – concurrency control.

TSQL Statement Types: DCL and TCL

Data Control Language DCL: which includes commands such as GRANT, and mostly concerned with rights, permissions and other controls of the database system.

DCL is the instruction set to maintain user access to a Relational DB and its Tables.

- GRANT – allow users access privileges to database.

REVOKE – withdraw users access privileges given by using the GRANT command.

Transaction Control Language TCL: which deals with transaction within a database.

TCL is the instruction set to maintain data integrity by allowing:

- COMMIT – commits a transaction.

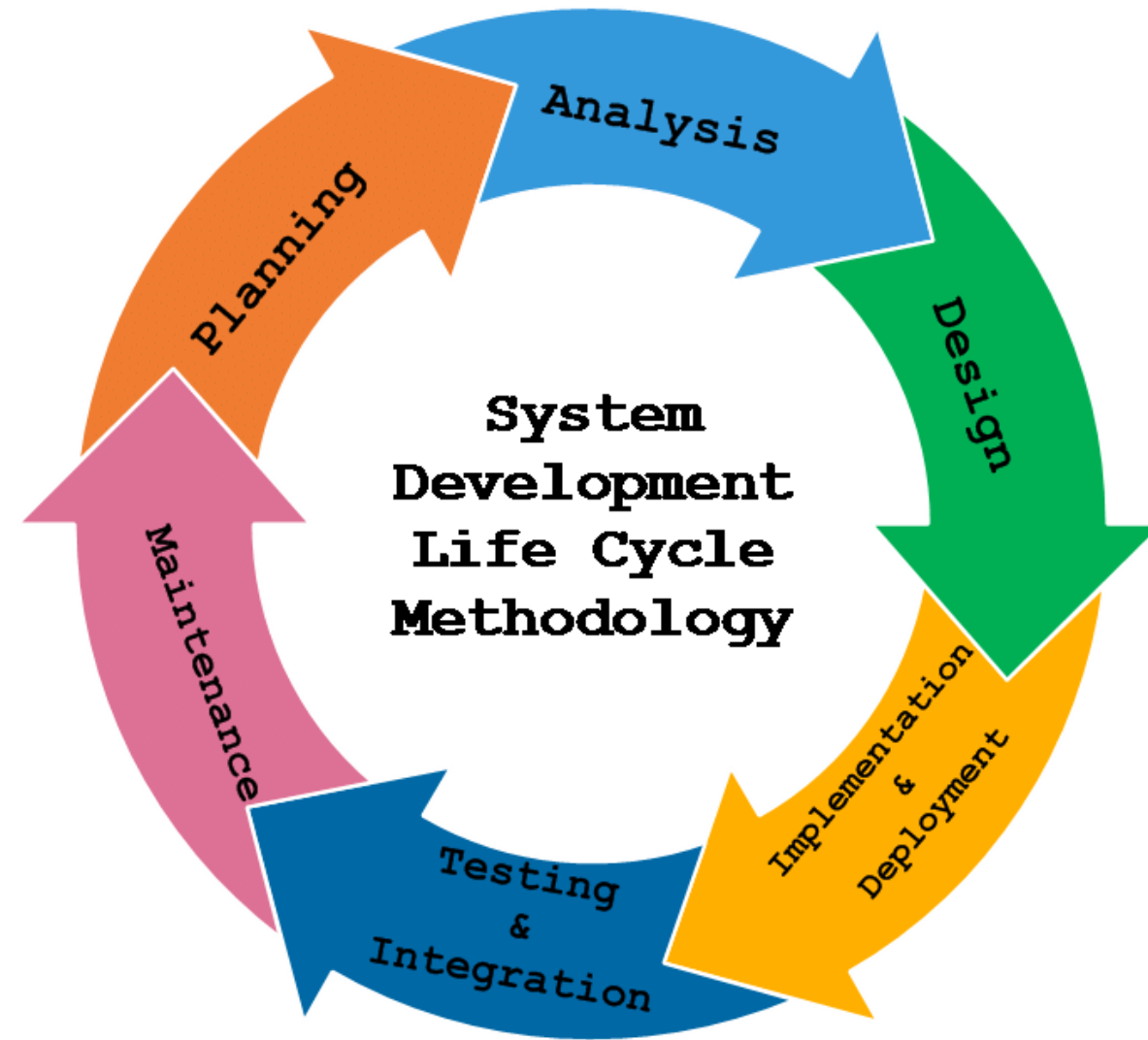
- ROLLBACK – rollback a transaction in case of any error occurs.

- SAVEPOINT – a point inside a transaction that allows rollback state to what it was at the time of the savepoint.

- SET TRANSACTION – specify characteristics for the transaction.

System Development Life Cycle (SDLC)

- The systems development life cycle (SDLC) is a conceptual model used in project management that describes the stages involved in an information system development project, from an initial feasibility study through maintenance of the completed application.
- SDLC can apply to technical and non-technical systems.

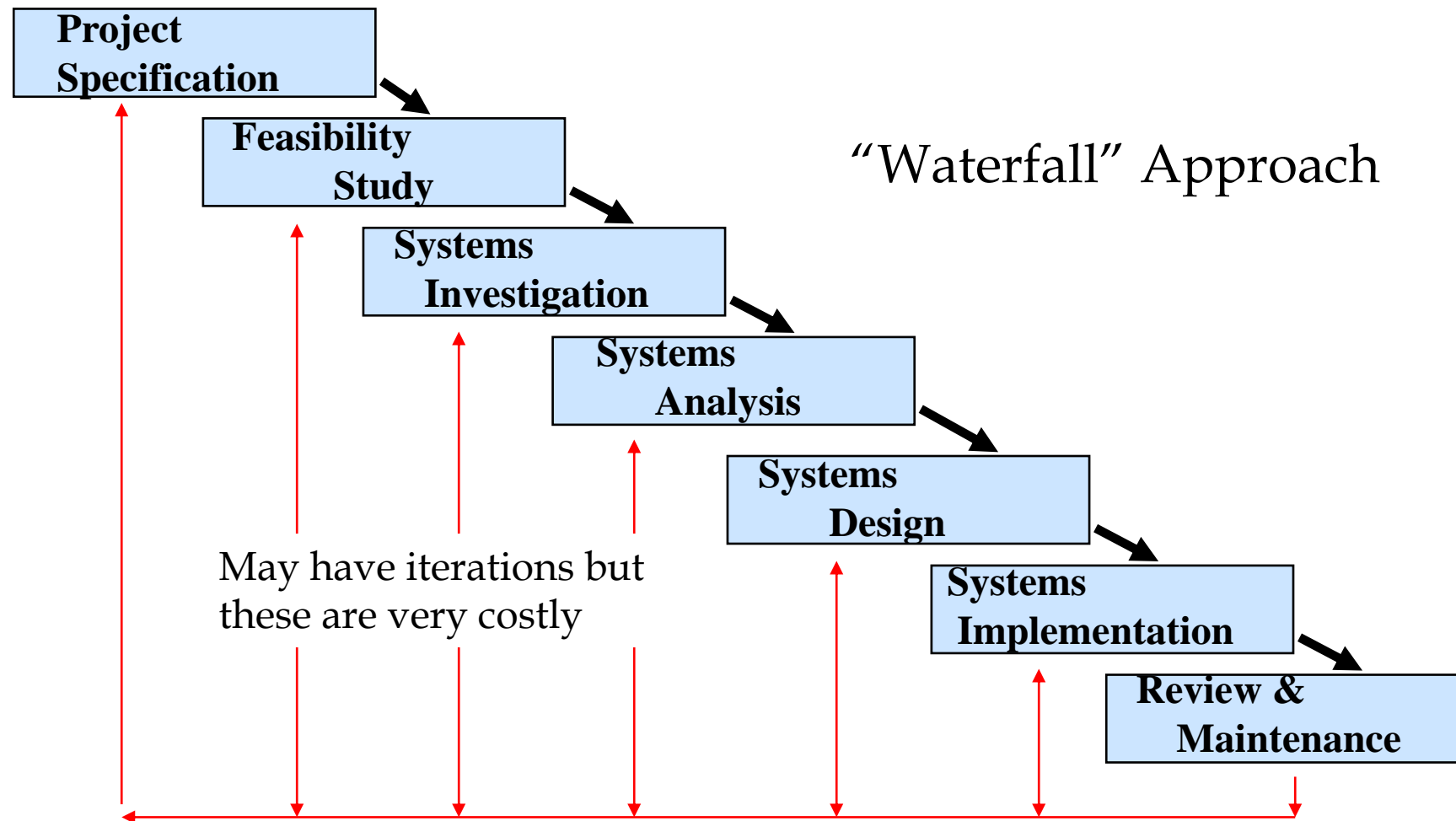


SDLC Methodologies

- Waterfall methodology
- V-model
- Iterative
- Agile methodology
- DevOps.
- Spiral
- ...

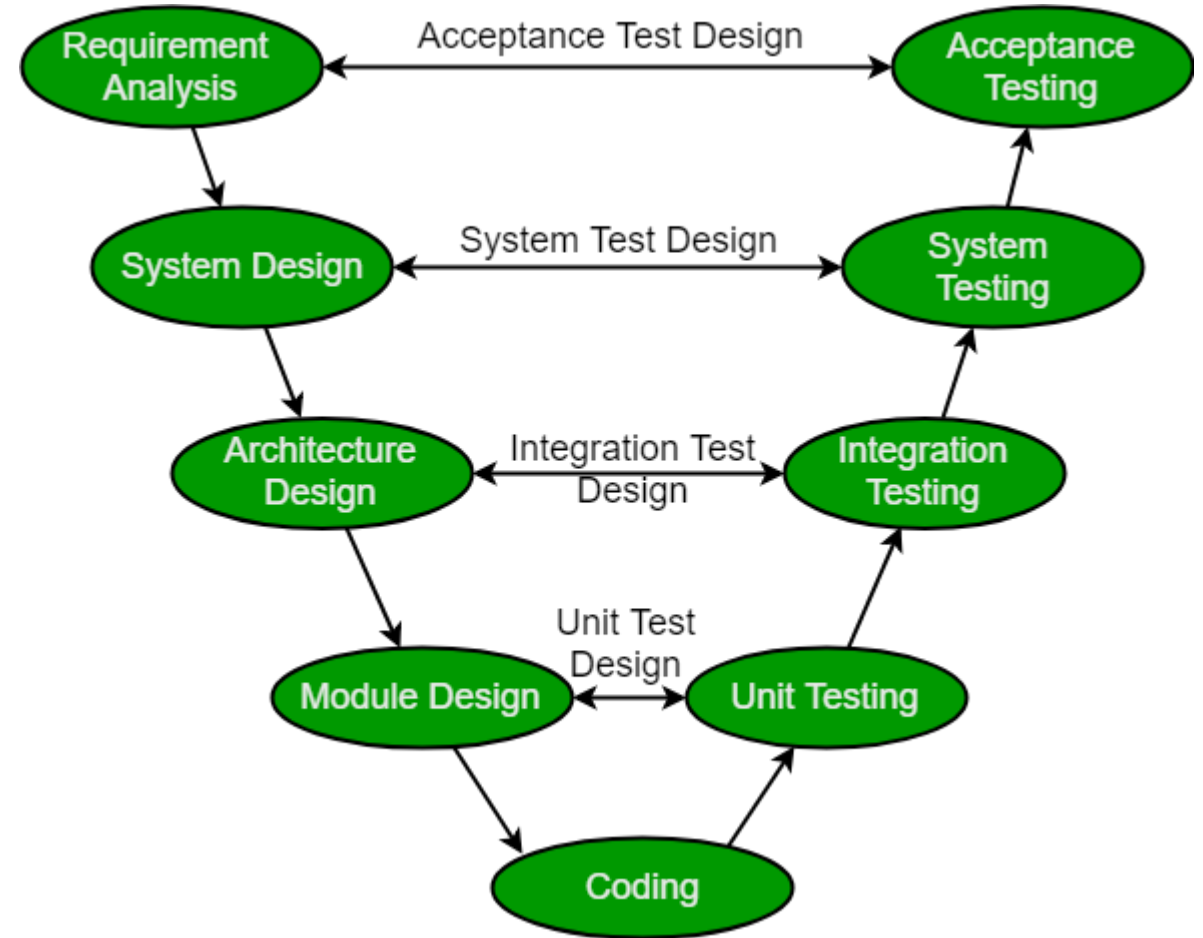
Waterfall Life Cycle Methodology

- The waterfall model is a sequential design process in which progress is seen as flowing steadily downwards (like a waterfall) through the phases



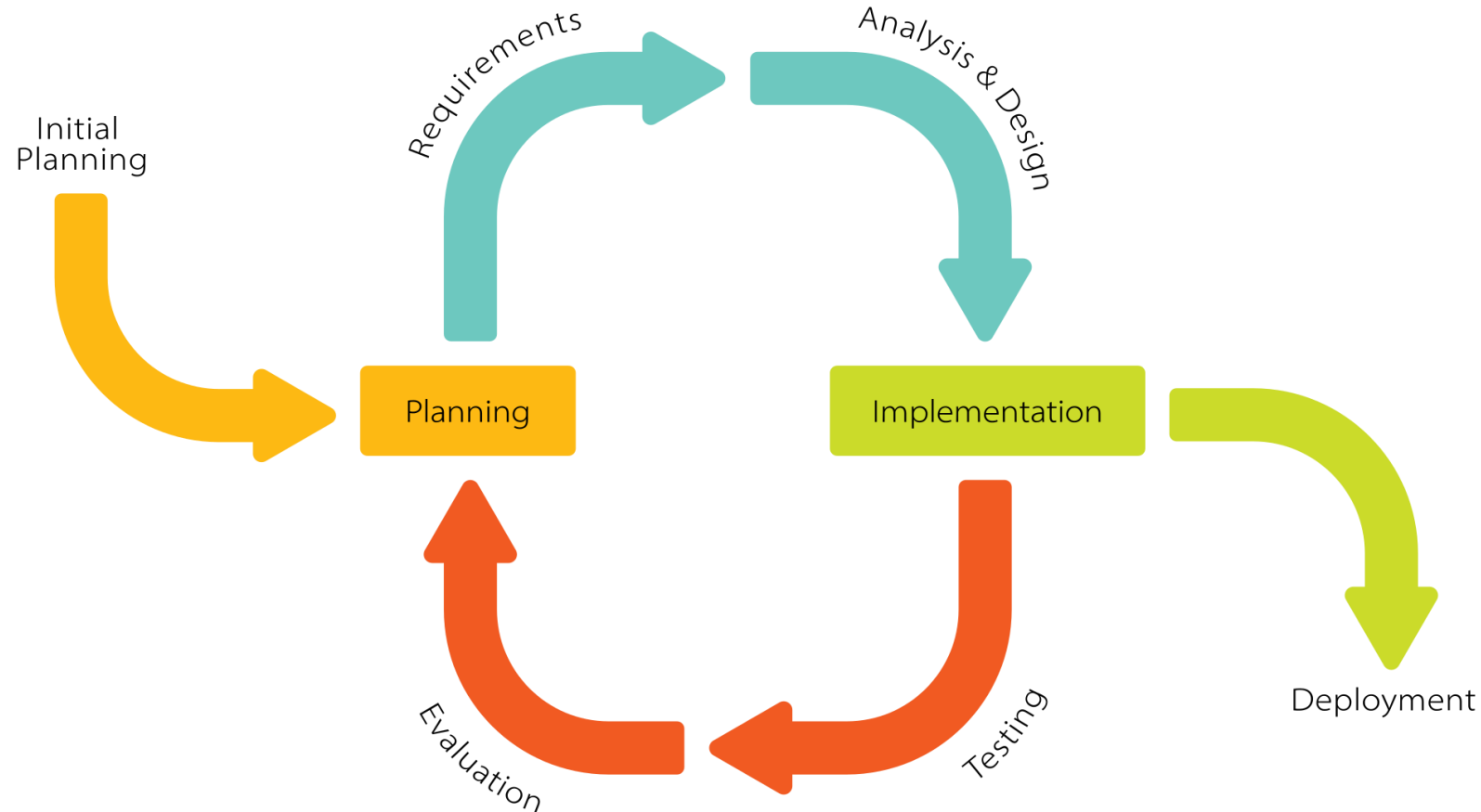
V-model Methodology

- The V-model is a type of SDLC model where process executes in a sequential manner in V-shape. It is also known as Verification and Validation model. It is based on the association of a testing phase for each corresponding development stage. Development of each step directly associated with the testing phase.



Iterative Methodology

- Iterative methodology allows software developers to adjust, refine, and review system development processes constantly to improve their performance incrementally.



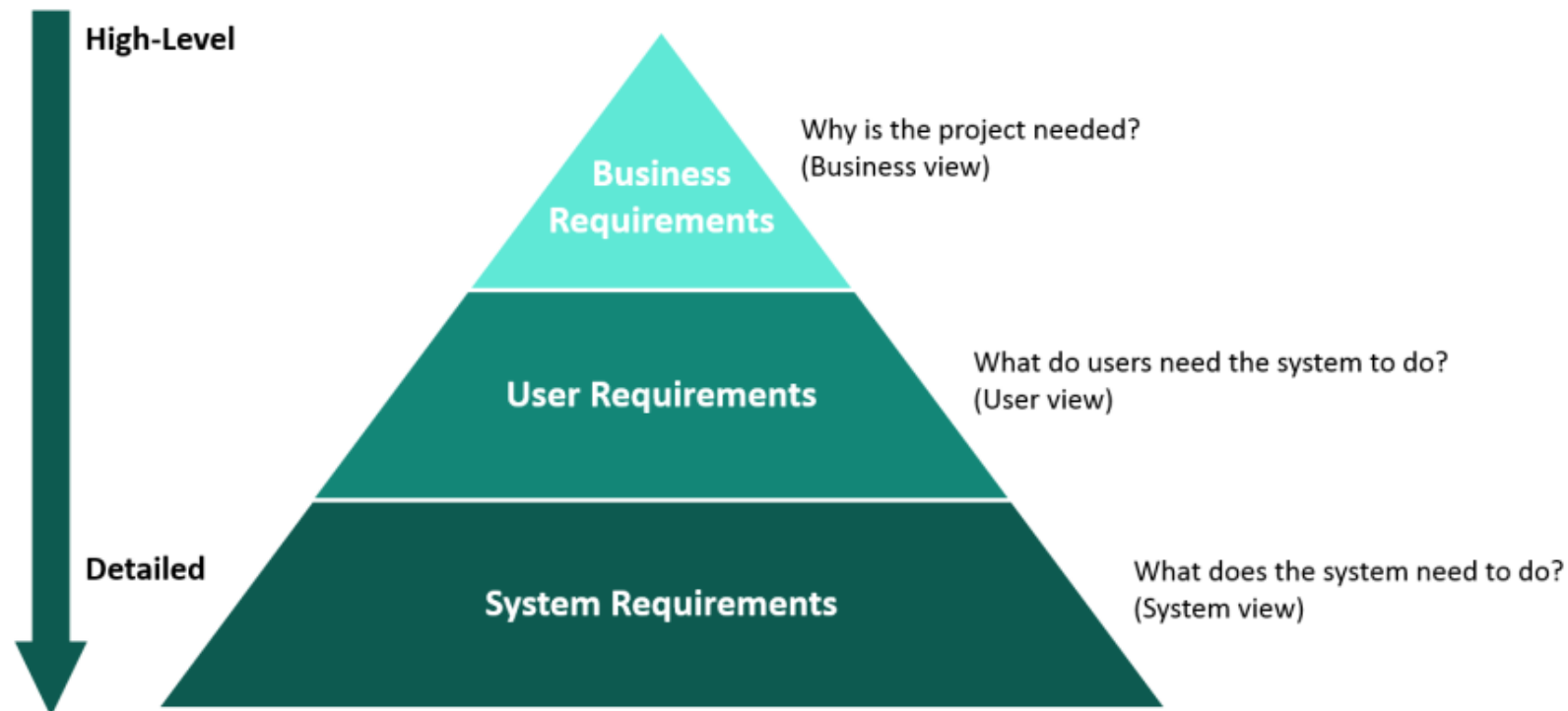
Agile Methodology

- The Agile methodology is a way to manage a project by breaking it up into several phases.
- It involves constant collaboration with stakeholders and continuous improvement at every stage.
- Once the work begins, teams cycle through a process of planning, executing, and evaluating.
- The agile iterative approach creates opportunities for constant evaluation and improvement in development processes.



System Requirements

- System requirements is a statement that identifies the functionality that is needed by a system in order to satisfy the customer's requirements.
- System requirements are a broad and also narrow subject that could be implemented to many items.



Requirements

System requirements is a statement that identifies the functionality that is needed by a system in order to satisfy the customer's requirements.

System requirements are a broad and also narrow subject that could be implemented to many items.

Functional Requirement

- Define what a product must do, what its features and functions are.
- Authentication of a user on trying to log in to the system

Non-Functional Requirement

- Describe the general properties of a system
- Testing such as usability, performance, stress, security, etc.
- The background colour of the screens should be light blue.



MoSCoW Method or MoSCoW Analysis

- It is a popular prioritization technique for managing requirements.
- The acronym MoSCoW represents four categories of initiatives:
 - must-have,
 - should-have,
 - could-have, and
 - won't-have, or will not have right now.



Demonstration – Requirement List & MoSCoW (Requirements List - workbook)

Sample Requirements List	
R1	The ECP shall display a list of all products offered by Music-Mansh Limited.
R2	The ECP shall organise the list of products by product category.
R3	The ECP shall display detailed product descriptions consisting of name, photograph, price and description on demand.
R4	The ECP shall display the number of items currently in the shopping basket on each page of the catalog.
R5	The ECP shall accept all major credit cards.
R6	The ECP shall validate payment with the credit card processing company.
R7	The ECP shall automatically calculate and add a shipping charge to the order.
R8	The ECP shall automatically calculate and add tax to the order.
R9	The ECP shall allow the customer to place items into a shopping basket.
R10	The ECP shall allow the customer to remove items from their shopping basket.
R11	The ECP shall allow the customer to “check out” and pay for their products.
R12	The ECP user interface shall support the insertion of adverts.
R13	The ECP shall raise a new customer order at the point of payment.
R14	The ECP shall notify the dispatch department once the order has been paid for.
R15	The ECP shall send a copy of the current order to the customer when payment is accepted and the order confirmed.
R16	The ECP shall allow customers to view their order history.
R17	The ECP shall allow a customer to cancel a customer order provided that the order has not been dispatched.
R18	The ECP shall notify the dispatch department when an order is cancelled.
R19	When the order is cancelled the ECP shall refund the cost of the order.
R20	The ECP shall allow the dispatch department to view all orders.
R21	The ECP shall allow a customer to register with the site.
R22	The ECP shall use the customer’s email address as the username for logon purposes.
R23	The ECP shall require the customer to set a password.
R24	The ECP shall collect customer information consisting of name, address, email address, phone number, fax number, credit card information.
R25	The ECP shall allow customers to view and edit their customer information.
R26	The ECP shall authenticate all customers prior to making payment, viewing outstanding orders or viewing customer information.
R27	The ECP shall support 100,000 transactions per day.
R28	The ECP shall support a peak transaction rate of 10 transactions per second.
R29	The ECP shall support 10,000 concurrent sessions.
R30	The ECP shall be available 24 hours per day, 360 days per year.
R31	The ECP shall store sales transaction data.
R32	The ECP shall accept payment and raise an order within 5 seconds in 95% of the cases.
R33	The ECP shall log in a customer within 5 seconds.
R34	The ECP shall use a browser as its user interface.
<div> Sample Requirements List Requirements List-Individual Requirements List-Group + </div>	

Requirement List - Example

Functional Requirements

ID	Details	Type	Priority
R0	The ECP shall accept all major credit cards.	<ul style="list-style-type: none">• Payment• Functional	<u>MustHave</u>
R1	The ECP user interface shall support the insertion of adverts.	<ul style="list-style-type: none">• User Interface• Functional	<u>WantToHave</u>

Non-Functional Requirements

R0	The ECP shall be available 24 hours per day, 360 days per year.	<ul style="list-style-type: none">• Availability• Non-Functional	<u>MustHave</u>
----	---	---	-----------------

User Story

- User story is an informal, natural language description of features of a software system.
- They are written from the perspective of an end user or user of a system, and may be recorded on index cards, Post-it notes, or digitally in project management software.

Writing a user story

- 1 Define your **end user**
- 2 Specify what **they want**
- 3 Describe **the benefit**
- 4 Add **acceptance criteria**

Copyright © 2019 Knowledge Train Limited

User Story

As a <role>
I want <goal>
so that <benefit>

Acceptance criteria:
(Conditions of Satisfaction)

...
...

As an Account Manager
I want a sales report of my account
to be sent to my inbox daily
So that I can monitor the sales
progress of my customer portfolio

Acceptance criteria:

1. The report is sent daily to my inbox
2. The report contains the following sales details: ...
3. The report is in csv format.

How to write User Story - Examples

- **As a Customer, I want** to provide credit card details for payment **So that** I can place an order.
- **As a User, I want** the system to support various features **So that** new adverts can be inserted easily in the future.
- **As an Order confirmation officer, I want** to notify the dispatch department once the order has been paid for **So that** it could be dispatched and delivered on time.

Writing a user story

- 1 Define your **end user**
- 2 Specify what **they want**
- 3 Describe **the benefit**
- 4 Add **acceptance criteria**

Copyright © 2019 Knowledge Train Limited

User Story

As a <role>
I want <goal>
so that <benefit>

Acceptance criteria:
(Conditions of Satisfaction)
...
...

As an Account Manager
I want a sales report of my account
to be sent to my inbox daily
So that I can monitor the sales
progress of my customer portfolio

Acceptance criteria:
1. The report is sent daily to my
inbox
2. The report contains the
following sales details: ...
3. The report is in csv format.

Supporting Material

- Highly recommended to walk through in your own time

System Development Life Cycle



<https://youtu.be/05Un7kPMibk>



<https://youtu.be/Xrgk023l4II>

Systems Design Phase in SDLC



<https://youtu.be/rAR5sbaphwU>

Software Development Lifecycle in 9 minutes!



<https://youtu.be/i-QyW8D3ei0>

HOW TO WRITE SOFTWARE REQUIREMENTS SPECIFICATION



<https://youtu.be/PtJmjPkrSUE>

What Is a Functional Requirement?



<https://youtu.be/wbCAiLzOE30>