THEORY PRACTICE CORRESPONDENCE

Hybrid Application for Warehouse Management System

Team members:

Chittoori Meha (S2)

Chaitanya Reddy (S2)

Challamala Krishna Priya (S2)

Jallepalli Prerna (S3)

Madarapu Srikar (S3)

Index

S.no	Contents	Page number
1.	SDLC and methodologies	2-15
2.	Requirement analysis and specification	16-20
3.	Software project management	21-27 and 30-32
4.	Techniques for project estimation	27-29
5.	Class , Use case , sequence diagram	33-37
6.	Software metrics	37-39
7.	System design	39-43
8.	Software engineering	43-46
9.	Testing	46-49
10.	Coding	49-58

1. Software development life cycle

Key concepts	Explore concepts' significance and relevance
• Software development life cycle is also referred as Application development life cycle • Application development life cycle is the term used in software engineering to describe the process of planning, creating, testing, and deploying an information system. Different phases of SDLC are: 1. Requirement gathering and analysis 2. Design 3. implementation or coding 4. Testing 5. Deployment 6. Maintenance • It is a framework that describes the activities performed at each stage of a software development project. • Different approaches to life cycle methodology are: waterfall model, Prototyping Model, Spiral Model, Incremental Model RAD model. • A life cycle model represents all the activities required to make a software product transit through its life cycle phases. It also captures the order in which these activities are to be undertaken.	• SDLC and its phases contribute to produce high quality system that meet or exceed customer expectations, based on customer requirements, by delivering the system which move through each clearly defined phase, within scheduled time and estimated cost. • The software development lifecycle is not only a great way to ensure your app meets the needs of your business and customers, but it is also essential in supporting the app once it's published. • By 2020, most SE course students will be working for an IT company. When working on a project, they must recall what kinds of SDLC's were most feasible to follow and analyses why their company chose the model that they did. It could be an organizational choice, which means they choose the model that they feel is most compatible with their culture and objectives. For example, originally developed by IBM, is the Cleanroom process in which each software increment is formally specified and this specification is transformed into an implementation. Software correctness is demonstrated using a formal approach. The main goal of the Cleanroom process is zero-defects software so that delivered systems have a high level of reliability. • Students should remember throughout the lifetime that before doing a task or a project he/she should go through the phases of SDLC which are mentioned to reach ones goal and expectation. • Before starting a project Planning helps us to estimate the cost and time and gives a view of steps to be followed in

	the entire project. Analysis phase is to determine where the problem is, in an attempt to fix the system. Design phase will describe the new system as a collection of modules or subsystems. Implementation phase is the main implementing (coding) part. After product completion taking care of the system is very important. By following all these phases it helps in our professional growth.
	• For example in sports a team must always plan their moves and estimate the strengths and weakness of the opponent team(planning) and analyzing team weakness(analysis) and should design a plan for it(design) and implement that on the field (implementation).

Establish relevance make sense and meaning	
Real life context	Find interdisciplinary connection
• The systems development life-cycle concept applies to a range of hardware and software configurations, as a system can be composed of hardware only, software only, or a combination of both. Following the assembly line manufacturing concept, an SDLC goals to produce high quality systems that meet customer expectations, based on customer requirements, by delivering systems which move through each clearly well-defined and planned phases, within fixed time schedules and estimated costs. • As Benjamin Franklin said, **"By failing to prepare, you are preparing to fail".** Setting goals and milestones beforehand or planning ahead increases the chances of success at	• For example a company has a goal to reach to that specified position in the market, the company then the company first notes it requirements and then analyze its strengthens and weakness and analyzes the same for their opponents. Then design a strategy to upgrade their position in the market and implement that strategy. These phases are not only used in technical activities but also it <u>interconnects</u> to many aspects in the society. • The SDLC process consists of six tailored phases that help manage a wide range of activity to conduct projects or automate House activities with information technology. SDLC is not limited to technical activity but it actually begins with customer needs

any given task. Since long term goals are the most difficult to achieve, people seek to short term goals, which will slowly but surely help them to move in the right direction towards their goal. SDLC helps us apply this idea to system development, giving the team working on the project a process, composed of a number of clearly defined and distinct work phases to adhere to. • This way, the team is organized and their actions well-planned. Even when the going gets tough, they do not lose focus of his goals and objective and have a well-drawn out plan to fall back on.	and evolves through different processes and user requirements to develop a solution to any problem. The primary objective of implementing a standardized SDLC policy is to provide coordinated excellent service, at low costs, to support the activity of customers and users. •

Establish relevance make sense and meaning	
Real life context	Find interdisciplinary connection
• The systems development life-cycle concept applies to a range of hardware and software configurations, as a system can be composed of hardware only, software only, or a combination of both. Following the assembly line manufacturing concept, an SDLC goals to produce high quality systems that meet customer expectations, based on customer requirements, by delivering systems which move through each clearly well-defined and planned phases, within fixed time schedules and estimated costs. • As Benjamin Franklin said, **"By failing to prepare, you are preparing to fail".** Setting goals and milestones beforehand or planning ahead	• For example a company has a goal to reach to that specified position in the market, the company then the company first notes it requirements and then analyze its strengthens and weakness and analyzes the same for their opponents. Then design a strategy to upgrade their position in the market and implement that strategy. These phases are not only used in technical activities but also it <u>interconnects</u> to many aspects in the society. • The SDLC process consists of six tailored phases that help manage a wide range of activity to conduct projects or automate House activities with information technology. SDLC is not limited to technical activity but it

increases the chances of success at any given task. Since long term goals are the most difficult to achieve, people seek to short term goals, which will slowly but surely help them to move in the right direction towards their goal. SDLC helps us apply this idea to system development, giving the team working on the project a process, composed of a number of clearly defined and distinct work phases to adhere to. • This way, the team is organized and their actions well-planned. Even when the going gets tough, they do not lose focus of his goals and objective and have a well-drawn out plan to fall back on.	actually begins with customer needs and evolves through different processes and user requirements to develop a solution to any problem. The primary objective of implementing a standardized SDLC policy is to provide coordinated excellent service, at low costs, to support the activity of customers and users. •

1.1 Classical Waterfall model

Key concepts	Explore concepts' significance and relevance
• The waterfall model is a non-iterative design process that is used in software development process, in which progress is seen as flowing steadily downwards. • Well known requirements are needed to implement this model, it is mainly used when we want to create a new version of existing product. • The different phases of waterfall model are: Software requirements phase, design phase, implementation phase, verification (testing) phase and operation (maintenance) phase. **Requirements:** this phase defines the information, behavior, performance and the interfaces of the system. **Design:** This includes system architecture, Algorithms details and interface representations. **Implementation:** this includes the source code, documentation **Test:** this phase is to check that all the modules are working or not and the whole system is meeting the specific expectations or not. **Maintenance:** this phase is for fixing the bugs and adding other functionalities.	• Waterfall model and its phases are important as they provide structure to in experienced staff and set requirement stability • From this concept Students should remember that in some cases requirements are to be very clearly identified and understood. • This concept is helpful in professional growth for a good management control (plan, staff, and track). • For example let us take real life situation that is an artist composing a song for a movie. He/she must know the specific requirement for the song (that is song played in particular situation) that cannot be changed for suppose and then design that song and then implement it and later test it.

Establish relevance make sense and meaning	
Real life contexts	Interdisciplinary connections
- This waterfall model plays a significant role mainly when the requirements of a customer are very well defined and will not change. This model is very easy to understand and provides a well-defined structure to the inexperienced staff satisfying the expectations of the customer. - It also sets requirement stability. - If a customer wants a new version of an existing product and when it is possible to produce a stable design then waterfall model is best to be used.	They are mentioned in real life context

1.2 Iterative model:

Key concepts	Explore concepts' significance and relevance
• An iterative process starts with a simple implementation of a part of software requirements and iteratively enhances the versions until the system is completed implemented. • At each iteration design modifications are made and functional capabilities are also added. • Phases of iterative model are: Design and development Testing Implementation Review phase.	• Its phases play a major role in generating working software quickly and early during the software life cycle. • Moreover its implementation cost is less to change scope and requirements. • By 2020 and for life time every student must remember that if the requirements of product are clear and want to identify regular bugs and mistakes in implementation of the project one should make use of this concept.

Establish relevance make sense and meaning	
Real life context	Find interdisciplinary connection
• **Benefits for customer**: it is easy to manage the risk as the risky pieces are identified and handled during early stages of implementation • Another real life advantage of this model is we can debug easily during a smaller iteration. • This model is very useful in real life being more flexible. • When it is time to the market constraint • Parallel development can be planned. • During life cycle software is produced early which facilitates customer evaluation and feedback.	• When we want to add new additional functionalities and want to identify the risks and cons at the early stage iterative model is best.

Engage in critical thinking	Technology, tools, techniques
- **Requirement phase**: in this phase requirements for the software are gathered and analyzed. - **Design phase:** In this phase software solution to meet the requirements is designed. This may be new design or new version of existing system. - **Implementation and test:** When the software is coded, integrated and tested. - **Review Phase:** In this phase software is evaluated, the current requirements and specifications are reviewed, and functionalities are added if required. - Major requirements must be defined. some functionalities may evolve with time.	

1.4 Spiral Model

Key concepts	Explore concepts' significance and relevance
- Spiral model is also called as a meta model - It's look like a spiral and has many loops, which are not fixed. Each loop represents a phase of the software process. - Each phase in this model is split into four : - First quadrant (Objective Setting) - Second Quadrant (Risk Assessment and Reduction) - Third Quadrant (Development and Validation) - Fourth Quadrant (Review and Planning)	- Its phases play a major role in Long-term project commitment Users are unsure of their needs - New product line - By 2020 and life time we should remember that if we have to do a project which need risk analysis one should use spiral model.

Establish relevance make sense and meaning	
Real life context	**Find interdisciplinary connection**
Benefits for customer: - When the project is large and mission critical project then it gives a lot of good and expected results to the customer. - When customer wants to add the Functionality at a later stage. - The spiral model development concept was presented on the example of e-materials for the respiratory physiology and pathophysiology study. - The spiral model development concept of multimedia application for the effective spiral evolution of IT and permanent improvement of the communication between a patient and the medical staff.	- This concept is used when software is produced early. - As mentioned in real life context section it also has interdisciplinary connection with medical industry.

1.5 Agile model(Model which is used)

Key concepts	Explore concepts' significance and relevance
• Agile SDLC model is a combination of iterative and incremental process models It focus on process adaptability and customer • Phases: Individuals and interactions over processes and tools Working software over comprehensive documentation Customer collaboration Responding to change	• This model and its phases are very important for the team to follow uniform rules • To complete the work in short periods of time. • By 2020 and life time, every student must remember that he/she whether want to complete a project in short time span then they should first divide the tasks and assign time schedules to them.

Establish relevance make sense and meaning	
Real life context	**Find interdisciplinary connection**
• **Benefits for customer**: • It is usually combination with extreme programming. • As following this models makes completely developed and tested in short iterations customer can get quick and decent satisfaction by team meeting the expectations of the customer. • Team follows uniform clearly defined rules which help the manager for managing the team and decreasing the disputes and misunderstandings with the customer. • Taking regular feedback from the customer may help to meet the expectations and increase the quality if the product. • **Real time usage**: Google is working on the project to come up with a best product for MS Word that provides all the features provided by MS Word and other features that are requested by the marketing team. The final product needs to be ready in 10 months of time.	• This concept is used when software is produced early. • Following this model and its phases means Self-organizing • Each team member carries a lot of responsibility. • Interdisciplinary connection: In marketing sector this concept is used, Each team member has responsibility of a task and should submit it in a scheduled time. • Not only in marketing but in also many fields there is an interdisciplinary connection with this concept.

Engage in critical thinking	Technology, tools, techniques
Phases: Individuals and interactions over processes and tools Working software over comprehensive documentation Customer collaboration Responding to change	

1.6 RAD Model

Key concepts	Explore concepts' significance and relevance
• RAD stands for Rapid application development. It emphasizes on an extremely short development cycle. • High-speed adaptation of the linear sequential model • Development time of the application is reduced compared to other models. • Phases: Business modeling Data modeling Process modeling Application generation Testing and turnover • Business modeling: flow of information is present among business functions • Data modeling: the information flow in above phase is defined into a set of data objects .The characteristics of each object are identified and defined. • Process modeling: Processing descriptions are created for adding, modifying, and retrieving a data object. • Application generation: It works as to reuse existing program components.	• This model and phases increases the reusability of the components. • But when technical risks are high it is not preferred to use this model • This model also encourages customer feedback • By 2020 and life time if anyone wants to make a less development time product one can use this concept of RAD model. • Can be used When a system is created which can be modularized in 2-3 months of time. • When there is high availability of designers and the budget is high

• Testing and turnover: It emphasizes reuse ,many of the program components have already been tested.	

Establish relevance make sense and meaning	
Real life context	**Find interdisciplinary connection**
Benefits for customer: - It emphasizes reuse; many of the program components have already been tested. This reduces overall testing time. New components must be tested - As it requires highly skilled developers and designers the chance of customer getting satisfied is more. - Real time Usage: - Getting started with the ideas of Barry Boehm and others, James Martin developed RAD approach during 1980's at IBM	- When a system is created which can be modularized in 2-3 months of time - When there is high availability of designers and the budget is high - Real time interdisciplinary connection is described in Real life context section.

2. Requirement Analysis and Specification

2.1 User Interfaces

Key concepts	Explore concepts' significance and relevance
- UI is an integrated part of any software as it describes the logical characteristics of each interface that the user encounters while using the software. - User acceptance is majorly dependent on how the user can use the software. - An accepted software has the following features :- - Easy to use - Interactive - Handling the operational errors effectively	- GUI Requirement Gathering 1. User Analysis 2. Task Analysis 3. GUI Design and implementation 4. Testing - Rules required for GUI designing 1. Strive for consistency 2 Enable frequent users to use shortcuts 3.Offer informative feedback 4. Design dialog to yield closure 5.Offer simple error handling 6 Permit easy reversal of actions 7. Support internal locus of control 8. Reduce short-term memory load

Establish relevance make sense and meaning	
Real life context	**Find interdisciplinary connection**
- In real world contexts, every web application or every web service that requires the interaction of the user with software for processing of data and retrieving data as per user. - Find Interdisciplinary connections: - user interfaces are used for any application that requires an interaction of user with the software, this involves any field in real	- User interface is the interface that is visible to the user. - User acceptance is majorly dependent on how the user can use the software. - Interdisciplinary connection is defined in real life context section.

world context, for instance, in the bank field, user interface interacts with the bank database for any modification of for retrieving data as per the users requirements.	

Engage in critical thinking	Technology, tools, techniques
- GUI Requirement Gathering: - this involves gathering of all the requirements that will consist for designing. - User Analysis: - this phase is analyzing the customer, for a technical customer, advanced and complex interface is provided and for a non-technical customer interface is provided accordingly. - Task Analysis: - tasks and sub-tasks for the software must be analyzed as it provides goals for GUI presentation. - GUI Design and implementation: -after dividing the tasks and into sub-tasks, then the code is implemented which is then self-tested by developers. - Testing: -it includes usability, compatibility, user acceptance and various other.	HTML CSS Bootstrap Java script

Plan project management	Project specification and project brief
•	• User interface in our project was a main component as it interacts with the user for its requirements of viewing reports for warehouse management or any further modification

2.2 Hardware interface

Key concepts	Explore concepts' significance and relevance
• It is basically the interconnection between two devices together. • Hardware interface is the interaction between the mobile interface with which the user interacts and the logical server which helps in processing The data and producing output according as per the needs. • Some of the factors which define the hardware interface are: - • The interaction between software and hardware • device types • nature of the data • communication protocols	For instance, a computer has some components: • OS • Hard disk • RAM • Keyboard • Mouse These components help in making users input to reach the CPU for processing and giving back a desired output.

Establish relevance make sense and meaning	
Real life context	**Find interdisciplinary connection**
• In real world contexts, any web application that the user interacts has got a back-end with which it interacts for processing the user's data, hence this interaction between the application i.e. the device interface, clicking events are called which are sent to server for further processing. • A very famous example is the computer, in which monitor is connected to CPU where the data after fetching from application will be processed and the resultant output will be sent back.	• Hardware interfaces are used anywhere which has its device interface connected to the backend, and it will tell us to which user interface it will further lead to.

Engage in critical thinking	Project specification and project sketch
• First message of user id and role goes to the business logic server the data regarding the authorization and authentication of the roles is then sent to database for checking the information, and if the data is present the will send the output sent by the business logic server to the interface for further interaction of data with user.	• As per our project, mobile interface is connected to business logic server which helps in returning next user interface for user as per the input provided.

2.3 Software interface

Key concepts	Project specification and project sketch
1. It can be defined by the method of communication between applications and between the hardware. 2. Factors that define are: - • Connection between the product and other software components that include databases, operating systems, libraries, tools etc. • Identification of the data items coming in and going out of the system.	• As per our project, When user as per his role enters his details in the login interface, where data is then sent to the business logic server for the authorization and authentication of the roles, logic is further then sent back as per the entered role for the further UI as in our application there will be 2 roles, Field Engineer and Product Manager. • In each case the UI is different, so will display the following pages accordingly. The information is further checked in with the ERP database for right details, once approved by it will redirect to other interfaces as per the way business logic server designs it. If not found in the database will redirect to another page i.e., Invalid User id.

3. Risk management

3.1 Software Requirements risk

Key concepts	Explore concept significance and relevance
1. This kind of risk must be identified during mentioning of user requirements. 2. Factors that define are: - i. Ambiguity requirements ii. Change of requirements in middle iii. Modification of requirements in middle	Various phases of risk management: - • Risk analysis • Risk Assignment • Risk Index

iv. Change extension of requirements	

Establish relevance make sense and meaning

Real life context	Find interdisciplinary connection
• In real world context, software requirement risks are very common as we may have change in requirements any time till the product is finished.	• Risk management is the most important issue involved in project development. Taking any project into consideration, risks must be identified and resolved during the development of project. And in case of software requirement risks, must be better understood and looked as product is based on requirements.

Engage in critical thinking

- Risk analysis: - there are different types of risk analysis that can be used. And this phase helps in identifying the level at which the risk is present in software project. It is declared to be the most important part in software design phase to evaluate criticality of the system, and ones these risks are found then various measures are taken to remove or to lower the level.
- Risk Index: - the two major factors that the risks are categorized are impact of risk events and probability of occurrence. Risk index can be categorized by low, medium, high on basis of product of occurrence and impact. It is very important for prioritizing of risks.
- Risk Assignment: - it is an important case which helps in integrating risk management and risk analysis. It requires correct explanations of target system and all security features.

3.2 Software costs risk

Key concepts	Explore concept significance and relevance
1. While estimating the values and if not done in the right manner then there may be various delays and effort put into it. 2. Factors that define are: - - Lack of estimation - Hardware issues - Human Error - Lack of good testing leading to errors in code - Lack of management cycle	Various phases of risk management: - - Risk analysis - Risk Assignment - Risk Index

Establish relevance make sense and meaning	
Real life context	**Find interdisciplinary connection**
• In real world contexts, estimating before beginning project, if done with any error will lead to huge difference in the outcome. Hence estimation of any project must be done adequately.	• Any error while estimating will lead to huge differences in outcomes so must be calculated in right manner. Risk management is the most important issue involved in project development. Taking any project into consideration, risks must be identified and resolved during the development of project.

Engage in critical thinking
- Risk analysis: - there are different types of risk analysis that can be used. And this phase helps in identifying the level at which the risk is present in software project. It is declared to be the most important part in software design phase to evaluate criticality of the system, and ones these risks are found then various measures are taken to remove or to lower the level.   - Risk Index: - the two major factors that the risks are categorized are impact of risk events and probability of occurrence. Risk index can be categorized by low, medium, high on basis of product of occurrence and impact. It is very important for prioritizing of risks.   - Risk Assignment: - it is an important case which helps in integrating risk management and risk analysis. It requires correct explanations of target system and all security features.

Project specification and sketch
We took this software cost risk into account before we executed the whole plan. It was very helpful for determining the risks.

3.3 Software quality risks

Key concepts	Explore concept significance and relevance

1. Though software gets built but there may be issues regarding the quality of result.
2. Optimized code must be used which gives the best results.
3. Must be tested on various tools.
4. Various factors are: -
    • Human errors
    • Lack of documentation
    • Poor understanding of requirements
    • Lack of standard in project
	Various phases of risk management: -
	    ▪ Risk analysis
	    ▪ Risk Assignment
	    ▪ Risk Index
	There must be efficient tools for testing the software to give the best results.

Establish relevance make sense and meaning	
Real life context	**Engage in critical thinking**
• In real world contexts, these risks are common if software is not tested in the right way. Hence efficient tools must be used upon which the software gets tested.	▪ Risk analysis: - there are different types of risk analysis that can be used. And this phase helps in identifying the level at which the risk is present in software project. It is declared to be the most important part in software design phase to evaluate criticality of the system, and ones these risks are found then various measures are taken to remove or to lower the level. ▪ Risk Index: - the two major factors that the risks are categorized are impact of risk events and probability of occurrence. Risk index can be categorized by low, medium, high on basis of product of occurrence and impact. It is very important for prioritizing of risks. ▪ Risk Assignment: - it is an important case which helps in integrating risk management and risk analysis. It requires correct explanations of target system and all security features.

3.4 Software scheduling risks

Key concepts	Explore concept significance and relevance

1. Scheduling risks arise when the short-term deadlines are not fulfilled to achieve long-term goal.
2. In other words, the task is not done as per the scheduled time.
3. Various factors defining are: -
    • Lack of required skill
    • Insufficient knowledge
    • Human errors
    • Modification of requirements
    • Lack of employment
	Various phases of risk management: -
	    ▪ Risk analysis
	    ▪ Risk Assignment
	    ▪ Risk Index

Establish relevance make sense and meaning	
Real life context	**Find interdisciplinary connection**
• In real world context, scheduling risks occur if there is poor communication between the team and lack of motivation to do, which will lead to delaying the deadline or schedule. Hence team should be well motivated to fulfill the task given.	• Motivation acts as a major factor in team to full fil any task, henceforth, taken any project team must be highly motivated to do the job given.

Engage in critical thinking
- Risk analysis: - there are different types of risk analysis that can be used. And this phase helps in identifying the level at which the risk is present in software project. It is declared to be the most important part in software design phase to evaluate criticality of the system, and ones these risks are found then various measures are taken to remove or to lower the level.   - Risk Index: - the two major factors that the risks are categorized are impact of risk events and probability of occurrence. Risk index can be categorized by low, medium, high on basis of product of occurrence and impact. It is very important for prioritizing of risks.   - Risk Assignment: - it is an important case which helps in integrating risk management and risk analysis. It requires correct explanations of target system and all security features.

3.5 Project estimation

Key concepts	Explore concept significance and relevance

1. For a project to stand out, it needs to be given an estimated measure of its various factors and be able to fulfil them.
2. Factors that define project estimation are: -
    ▪ Size
    ▪ Time
    ▪ Effort
    ▪ Cost
	Various phases of Project estimation are: -
	▪ Software Size estimation
	▪ Software time estimation
	▪ Software cost estimation
	▪ Software effort estimation

Establish relevance make sense and meaning	
Real life context	**Find interdisciplinary connection**
• In real world scenarios, estimation plays a major role. As correct estimation, can control and manage the project more efficiently and effectively.	• This concept of estimation is considered in all the professional project for effective and efficient results. One of the tools that is being used most often is the concept of Work Breakthrough Structure (WBS) in which the work is divided into smaller goals and each one is then estimated and integrated to find the final estimated of the overall project.

Engage in critical thinking
o <u>Software Size estimation</u>: - size estimation can be done in terms of Kilo Line of Code. Lines of code depends on coding practices. o <u>Software effort estimation</u>: - it is the amount of estimated effort put into the software as per estimated size of software. It is usually done by organizations' effort history and by using certain standard formulas. o <u>Software time estimation</u>: - time can be estimated once the above two are. It is the sum of all the estimated time of sub tasks which gives us the final projects' estimated time. o <u>Software cost estimation</u>: - It depends on hardware, travel involved, tools and licenses, etc.

3.6 Project scheduling

Project purpose

Key concepts	Explore concept significance and its relevance
- Software project like any other projects main aim is to fulfill the purpose of project. - Factors that go defining project purpose are: - Project Objectives - Business Objectives - Other phases of this module are: - - Meet User Requirements - Meet Schedule deadlines - Be within budget - Produce Quality Results	For any project's purpose to me fulfilled it must undergo certain phases and some of them are: - - Meet User Requirements - Meet Schedule deadlines - Be within budget - Produce Quality Results

Establish relevance make sense and meaning	
Real life context	**Find interdisciplinary connection**
In real world context, any project has whether that be a chemistry project or software engineering itself, project purpose which defines objective of project and can work in a systematic manner.	Project Purpose is such that it is a must for any project that is going to be developed, not only a software but for any topic.

Engage in critical thinking
Phases: --   - Meet User Requirements: - the main objective of any project is to fulfil requirements of the customer to whom the product is going to be sold, or the customer who is going to use the product.   - Meet Schedule deadlines: - Software to be submitted on time must divide the objective into smaller tasks and fulfil every task as per the schedule deadlines which will eventually help in meeting deadlines of the main objective to be fulfilled.

- **Produce Quality Results**: - The project must produce desired and quality output avoiding any errors and having the best error handling techniques. The project must be tested on best tools for best quality results.

Project specification and project sketch

In our project, user requirements were kept in mind all through while making the product and any changes to be made were first tested with best tools as per the language which helped in finding out errors while integration of different modules or components of software. Project Purpose was fulfilled.

Project scope

Key concepts	Explore concepts' significance and relevance
- Scope of the project provides a detailed description of features functions, constraints, interfaces of software. - Functions are tasks the software is expected to perform and features are referred to as the attributes required in software as per user requirements, and constraints are referred to as the limitations imposed on software by hardware. - Factors that define Project scope: - - Elements that are included and excluded in the project - Various Processes and entities - Functions and features as per user requirements .	Phases that project scope have: - - Meet User Requirements - Identification of cost estimation - Identification of risks

Establish relevance make sense and meaning

Real life context	Find interdisciplinary connection
In real world scenario, project scope will let us know what is our project up to and where does it stand in front of the rest of the already existing projects. In other words, features, functions of project will differentiate from other existing and similar projects.	Different projects having various different features and functions

- Meet User Requirements: - the main objective of any project is to fulfil requirements of the customer to whom the product is going to be sold, or the customer who is going to use the product.
- Identification of cost estimation: - Having an estimated view on time and effort needed to complete the project will help in setting the deadlines in a right manner and fulfilling them accordingly. Cost in
- Identification of risks: - Risks can be of various forms, technical risks, business risks, hence which have an adverse effect on project schedule and hence increases the cost of the project. Hence, identification of certain common risks that may occur will help in estimation of cost of the project.

Project planning

Key concepts	Explore concept significance and its relevance
- It is a main component, as this describes how systematically a project can be taken forward. - Various Factors that define are: - - Project cost - Risks - Project Requirements	Phases that define: - - Identification of Project Requirements - Identification of cost estimation - Identification of risks - Project Plan

Establish relevance make sense and meaning	
Real life context	**Find interdisciplinary connection**
In real world contexts, projects must be planned in a systematic fashion to fulfil the objective and produce output on desired date.	Project planning is a very important task that a project manager does, after viewing project plan, sets deadline and divides the goal into smaller goals and gives them each a deadline to fulfil. This is applicable to any project.

Engage in critical thinking
- Identification of Project Requirements: - Before starting any project, having identified the project or the user requirements will help take the further activities in a systematic manner.   - Identification of cost estimation: - Having an estimated view on time and effort needed to complete the project will help in setting the deadlines in a right manner and fulfilling them accordingly. Cost in   - Identification of risks: - Risks can be of various forms, technical risks, business risks, hence which have an adverse effect on project schedule and hence increases the cost of the project. Hence, identification of certain common risks that may occur will help in estimation of cost of the project.   - Project Plan: - once all the above are identified in the desired fashion, project plan and deadline date can be set.

Project specification and project sketch
In our project, user requirements were kept in mind all through while making the product and any changes to be made were first tested with best tools as per the language which helped in finding out errors while integration of different modules or components of software. Project plan was fulfilled

4. Techniques for project estimation

4.1 Decomposition technique

Key concepts	Explore concept significance and relevance
1. For a project to stand out, it needs to be given an estimated measure of its various factors and be able to fulfil them. 2. Factors that define project estimation are: - - Size - Time - Effort - Cost	Models that are included: - - Line of Code - Function Points

Establish relevance make sense and meaning	
Real life context	**Find interdisciplinary connection**
• Estimation of size of project helps setting the deadline and finishing the task as per it. As correct estimation, can control and manage the project more efficiently and effectively.	• This concept of estimation is considered in all the professional project for effective and efficient results. One of the tools that is being used most often is the concept of Work Breakthrough Structure (WBS) in which the work is divided into smaller goals and each one is then estimated and integrated to find the final estimated of the overall project.

Engage in critical thinking

- Line of Code: - To begin the project, it is important to determine the size of it, since project size helps in estimation of various measures.
- Function Point: - Estimation is done keeping in mind the no. of function points present in the system.

Project specification and project sketch

In our project, before beginning it, we estimated the Line of Code to be at around 8,000 lines which includes the front-end, server and the database, with the connection.

4.2 Empirical technique

Key concepts	Interdisciplinary connections
1. This estimation is done using standard empirical formulae which are derived from Line of Code and Function Points. 2. Models that are present under this are: - COCOMO: - COnstructive COst MOdel	• This concept of estimation is considered in all the professional project for effective and efficient results. One of the tools that is being used most often is the concept of Work Breakthrough Structure (WBS) in which the work is divided into smaller goals and each one is then estimated and integrated to find the final estimated of the overall project.

Engage in critical thinking
• It was discovered and developed by Barry W. Boehm, it divides software product into 3 categories organic, semi-detached, and embedded.

Project specification and project sketch
We did not use this technique in our project for estimation.

5.1 Class diagram

Key concepts	Explore concepts' significance and relevance
• In software engineering a class diagram in the unified modeling language (UML) is a type of static diagram which describes the structure of a system by showing the system classes, their attributes, operations and also relationship among the objects. • The class diagram is the main building Block of object oriented modeling. • The class diagram just like sequence diagram gives sequence of flow of application but a little bit different.	• Class diagram is used for analysis and design of the static view of the application. • By 2020 and life time we should remember that before coding an application one should design first and class diagram is one of the main component of the design.

Establish relevance make sense and meaning	
Real life context	**Technology, tools, techniques**
• It is useful to describe responsibilities of a system. • It is the base for component and deployment diagrams. • It is used for forward and reverse engineering. • Class diagrams are also used in design document to represent the each class (module) of the system.	• Tools that are used for Class diagram that are used in our project are Argo UML and creately.

Plan project management	Project specification and project brief
•	• Class diagram in our project was one of the main component in design phase which guided us toward the right implementation of the code

5.2 Use case diagram

Key concepts	Explore concepts' significance and relevance
• Use case diagrams are usually referred as behavior diagrams used to describe set of actions that a system should perform in association with the external users. • A use case diagram is a representation of user's interaction with the system and shows the relationship between users and different use cases in which users are involved. • A use case diagram can identify the different type of users of a system and different use cases.	• Use case diagrams are used to specify: 1. Requirements, required usages of a system under design or analysis to capture that what system is supposed to do. 2. What system can do; 3. By defining how environment should interact with the subject so that it will be able to perform its services. • By 2020 and life time every student should remember that use case diagram helps to provide higher level view of the system.

Establish relevance make sense and meaning	
Real life context	**Technology, tools, techniques**
• The real life time purpose of Use case diagram is to capture the dynamic aspect of a system. • It is used to get an outside view of a system. • It is used to identify the external and internal factors influencing the system and also to show the interactions among the requirements are actors.	• Tools that are used for Use case diagram that are used in our project are Argo UML and creately.

Plan project management	**Project specification and project brief**
•	• Use case diagram in our project was one of the main component in Requirement phase which guided us toward the right implementation of the code. • It is used to gather requirements of a system.

5.3 Sequence diagram

Key concepts	**Explore concepts' significance and relevance**
• A sequence diagram is an interaction diagram that shows how objects interact with one another in what order. It is a construct of a message sequence chart. • Sequence diagram are also called as event diagrams. • The sequence diagram models the collaboration of objects based on time sequence.	• Sequence diagram is used to show the interactions between the objects in the sequential order that interactions occur. • Each key collaboration, diagrams are created to show how objects interact in various representative scenarios for the collaboration. • By 2020 and life time every student must remember before implementing the code one should clearly know the order of interaction of the objects for an effective design.

Establish relevance make sense and meaning	
Real life context	**Technology, tools, techniques**
- The real life time purpose of sequence diagram is to capture the dynamic aspect of a system. - It is used to get an order of interaction of objects.	Tools that are used for Use case diagram that are used in our project are Argo UML and creately.

Plan project management	Project specification and project brief
	- Use case diagram in our project was one of the main components in Design phase which guided us toward the right implementation of the code. - It is used to view the order the objects of a system.

6. Software metrics

Key concepts	Explore concepts' significance and relevance
• Software metric is a measure of degree to which a software system possesses some property. • They are two type of measures: • External measure: Cost, Effort, LOC, Speed and memory • Internal measure Functionality, Quality, Complexity, Efficiency, Reliability, maintainability • Different type of metrics are : • Complexity metrics • Halstead metrics • McCabe's complexity metric • Cyclomatic complexity • Design metrics	• Software metrics significance is to identify the following things: 1) How many defects exist within this given module? 2) How many test cases are executed per person? 3) What is test coverage? • To know schedule of the project • Workload and resource usage • They set the expectations • If the metrics are not done properly then assessment of software quality is purely subjective which arises to disputes at the end of development life cycle.

Establish relevance make sense and meaning	
Real life context	**Technology, tools, techniques**
• Benefits of customer: generation of software metrics is very important responsibility for Software test lead to take the next decisions for scheduled costs and time of activities. Understanding the kind of improvement needed to improve project quality to meet or exceed the customer expectations and satisfying them. It is also used by them for risk analysis of the product.	C#(1.2,2.0,3.0,4.0,5 and 6) COBOL (IBM Enterprise) Java 1.4-1.8 Logix5000(Iever 2.6) VBscript (Beta)

Plan project management	Project specification and project brief
•	• Software metrics in our project plays a major role for taking decisions for scheduled costs and time of activities. • Understanding the kind of improvement needed to improve project quality to meet customer expectation.

Engage critical thinking

- **Halstead metrics**: Program length: $N = N_1 + N_2$

- Program vocabulary: $n = n_1 + n_2$

- Estimated length: $= n_1 \log_2 n_1 + n_2 \log_2 n_2$

    - Close estimate of length for well structured programs

- Purity ratio: PR =   /N

- Volume: $V = N \log_2 n$

- Number of bits to provide a unique designator for each of the n items in the program vocabulary.

- Difficulty

- Program effort: $E = D*V$

- This is a good measure of program understandability

McCabe's complexity metric:

Nodes represent the tasks and edges represent the direction of flow between them.

Cyclomatic complexity:

- $V(G) = E - N + 2$

    - E is the number of flow graph edges

    - N is the number of nodes

    - $V(G) = P + 1$

    - P is the number of predicate nodes

7. System design

KEY CONCEPTS	CONCEPTS SIGNIFICANCE AND RELEVANCE
System design is the process of transforming user requirements to a suitable form, in the process we would define the architecture, the components, the modules, interfaces, and the data for a system to satisfy user's specified requirements. There are a couple of variants of system designing : 1) Structured design - Cohesion - coupling 2) Function oriented design 3) Object oriented design - Objects - Classes - Encapsulation - Inheritance - Polymorphism 4) Logical design - Entity relationship diagrams 5) Physical design - U/I design - Data design - Process design There are to approaches to system designing: 1) <u>Top down approach</u> In this approach the whole software system is considered as a single entity and is fragmented into sub systems based on various characteristics. 2) <u>Bottom up approach</u> In this approach the system starts from scratch considering the smallest of the components and building more complex and higher components with the use of smaller ones. Some of the really important concepts that come in handy while developing a system under the system design phase are: 1) Abstraction 2) Modularity, coupling and cohesion	It is the first step of the Software Design Life Cycle. It moves the concentration from the problem domain to the solution domain. It specifies how to fulfil the requirements mentioned in SRS. System design involves identifying the data sources, the nature and the type of data that is available. This helps in understanding what kind of data is available and by whom is it being supplied to the system in order for the system to be designed considering all the relevant factors. It should fulfil the needs of the users and the system should be user oriented/friendly. Things we learnt in 2015 which are important for future references , and these are applicable for both 2020/2050: 1) <u>Abstraction</u> Its main aim is to find the hierarchy among the data. This is valid throughout life as designing of the system will be based on the user requirements and which data facilitates these requirements the most. 2) <u>Information hiding</u> It will help to make better systems as it reduces access of information to users where there is potential of threat or damage to the system. Information hiding is strongly related to abstraction. 3) <u>Software science formulae</u> The various mathematical formulae that will be used remain evergreen as they are the basis to the system design phase process. 4) <u>Coupling and cohesion</u> <u>Applications around the world</u> • Benchmarking –evaluating how current systems perform • Computer programming and debugging in the software world, or detailed design in the consumer, enterprise/ commercial world – it specifies the final system of components. • Design – designers produce various models of a system or project on the basis of their analysis of

	the components. A document is produced that has a description of the system, but nothing is specific
3) Information hiding 4) Limit complexity 5) Hierarchical structure	• Requirements analysis – analyzes the needs of the customers. • Systems architecture – creates the blueprint or the model of the design that contains all the necessary components. • System testing – evaluates actual functionality to expected functionality, including all the various integration aspects.

ESTABLISH SENSE AND MEANING

REAL LIFE CONTEXT	INTERDISCPLINARY CONNECTIONS
Used in various industries like : 1) Hospital and pharmaceutical companies 2) Big organizations eg:- coca cola , google, etc. 3) Fitness industry 4) Advertisements and movie / animation industry. 5) Aviation industry.	System design comes in the smallest of the process such as forming of a great leadership. In this there is a choosing party that puts up its requests and requirements of the type of people it would need to head an organization. These heads that make the base further widen the organization (basic components lead to complex and bigger ones). They then choose the managing committee which happens to work under them and is bigger to them. They are directly related to the leaders. The managing committee then selects the working committee which works for the organization to build it up which is the largest group out of all. This way a proper system is framed which works together and is governed by a set of rules and regulations.

ENGAGE IN CRITICAL THINKING	TECHNIQUES, TOOLS, TECHNOLOGY
There are 2 major approaches to system designing process: Top down and bottom up approach. Both the approaches are used in various fields of work. They have their own pros and cons that come with using these approaches. Top-down approach **Advantages**: Organization will realize a focused use of the resources by individually managing application. The first implementation is a showcase for the management of identity. A much deeper and more mature implementation of the identity management solution takes place.	

Operation and maintenance resources are not impacted as badly as in the case with the bottom-up.

Disadvantages:

Limited coverage in the initial phases.
A minimum percentage of the user accounts will be managed in the initial phases.
Development of custom adapters at an initial stage.
Benefit of the solution is not recognized rapidly.
The implementation cost is high.
Bottom-up approach

Advantages:

Benefits realized in early phases.
Early automation replaces manual processes.
Implementation of password management for a great number of users.
Development of custom adapters at early stage is not likely to happen in bottom up.

Disadvantages:

The organizational structure you might change in a later stage
This strategy is driven through the existing infrastructure instead of our business processes.

We prefer on using a bottom up approach because, it is the joining together of systems to give rise to more complex systems, hence making the previous systems part of the emerging new systems. Bottom up takes very detailed care of each and every component and examines and works on the resources available to make a proper system that matches the user's requirements and meets his expectations. Through this approach we can manage our errors in a better manner and also have a wide variety of variations in the final model that the developers invent.

PLAN PROJECT MANAGEMENT	PROJECT SPECIFICATION AND PROJECT BREIF
INTERMEDIATE We can have a demo working model where the system designed is not perfect but may have some more improvements left. The intermediate system will have taken in consideration most of the user requirements and would have worked upon its resources to meet the basic working model. FINAL We have a fully functional model where all the user requirements and expectations are met, the final model can be approachable both through top down and bottom up approaches. Abstraction and information hiding are implemented for proper control and monitoring of the system that's been designed.	Our project is the warehouse management system. We have various entities that are part of the system that are interconnected and interrelated to one another.

8. Software engineering

Key concepts	Explore concepts' significance and relevance
• Software engineering is an engineering branch associated with development of software product using well-defined principles, methods and procedures. • The outcome of software engineering is an efficient and reliable software product. • Phases of software engineering are: • Introduction • Analysis phase • Design phase • Implementation phase • Testing phase • Introduction	• Software engineering and its phases are very important to build reliable software within scheduled time and budget with minimized errors and bugs. • Software engineering ensures that a system will be operational transitional and maintained. • By 2020 and life time every student must remember that project is divided into phases before implementing it. This gives better results.

- The Analysis Phase:

    - Things
    - Actions
    - States
    - Typical Scenarios
    - Atypical Scenarios
    - Incomplete and Non-Monotonic Requirements

- The Design Phase

    - Architecture

    - Implementation Plan

    - Critical Priority Analysis

    - Performance Analysis

    - Test Plan

- The Implementation Phase

    - Critical Error Removal

- The Testing Phase
- Regression Test
- Internal Testing
- Unit Testing
- Application Testing
- Stress Testing

Establish relevance make sense and meaning	
Real life context	Find interdisciplinary connection
- They are many software types: - System software, real time software, business software, PC software, AI software and son on. **Benefit to customer:** - As the team follows particular phases customer will be able to know the scheduled time of the product and will be satisfied at most.	- They are interdisciplinary connections to many fields as mentioned in the real time context section.

8.1 Software Validation

Key concepts	Explore concept significance and relevance
- Validation is process of checking whether the software satisfies the user requirements or not. - It is validated, if the software matches the requirement of the user. - It is done at the end of the SDLC.	- Validation answers the question – "Are we developing the product which attempts all that user needs from this software?" - By 2020 and lifetime student should remember that validation of a product is very important to check whether it satisfies the user requirements or not.

Establish relevance and make sense and meaning	
Real life context	**Interdisciplinary connections**
1. If during verification some defects are missed then it can be caught in validation process as failure 2. Validation is done during testing like feature testing, integration testing, system testing, load testing, compatibility testing, stress testing, etc. 3. It makes the product more user friendly.	In any field verification is required to avoid the defects so this concept of software validation is very useful in our real life.

9. Software testing

Key concepts	Explore concept significance and relevance
• It confirms whether the software meets the business requirements, and is developed as per the proper design specifications. **Target of the test are -** • **Errors** - These are the coding mistakes which are made by developers. • **Fault** - A fault or bug is a result of an error and can cause system to fail. • **Failure** – when the system is not able to perform the desired task.	• Verification answers the question– "Are we developing this product by firmly following all design specifications?" • By 2020 and life time one should remember that to confirm that software meets the requirements and is developed according to the proper design, then we should apply the concept of software testing.

Establish relevance and make sense and meaning	
Real life context	**Interdisciplinary connections**
1. It helps in reducing the number of defect in the later stages of development. 2. reduces the chances of failures 3. provides the customer a reailable product.	In any field there is a necessity to verify the final system/product to produce a reliable product to the customer.

Tools, Technologies and techniques	Project specification and project brief
IcuTest QF-Test Ranorex and many more	We used testing in our project to detect the bigs and errors it played a crucial role in making our product complete.

Types of Testing

Key concepts	Explore significance of the concept and relevance
- Functionality testing It is performed to check whether the software application and functions behave correctly according to design specifications. - Black-box testing It is also called 'Behavioral' testing. In this only the functionality is being tested without taking care of the actual implementation - White-box testing It is conducted to test program and its implementation. It is also known as 'Structural' testing.	- Black box testing increases the chance of reproducibility. - The environment where program is been executed is also tested. - Tester needs no knowledge for implementing this not even programming languages .White box testing helps to increase the codes efficiency and structure - By 2020 and life time student if he want to increase the efficiency of the program or want to reproduce the components he/she must use white box and black box respectively

Engage critical thinking	Tools, techniques and technologies
Black-box testing techniques: **Equivalence class** - The input is divided into same classes. If one element of a class passes the test, it is assumed that all the elements of the class is passed. **Boundary values** - The input is divided into higher and lower end values. If these values pass the test, it is assumed that all values in between may pass too.	

- **Pair-wise Testing** - In this multiple parameters are tested pair-wise for their different values.

- **State-based testing** - The system changes state on provision of input. These systems are tested based on their states and input.

White-box testing techniques:

- **Control-flow testing** – it's purpose is to set up test cases. Test cases covers all statements and branch conditions.

- **Data-flow testing** – it cover all the data variables included in the program. It tells and tests where the variables are been declared, defined and where they were used.

The following is needed to be checked during the functionality testing:

- Installation and setup on localized machines
- Text input
- Core application functions.
- Text handling (such as copying, pasting, and editing) of extended characters, special fonts

Establish relevance and make sense and meaning	
Real life context	**Project specification and project brief**
In real life if someone wants to increase the efficiency and make a better design they may use white box testing and in order to increase the reproducibility they can use black box testing.	Black box testing white box testing and functional testing plays an important role in our project.

Testing levels

Key concepts	Explore significance of concept and its relevance
- Testing is done at various levels of SDLC. The testing process runs parallel to software development. Before going to the next stage, a stage is tested, validated and verified. - Unit Testing Integration Testing System Testing Performance testing – Comparison Testing Limit Testing Stress Testing Security testing Usability testing Documentation testing Acceptance testing Regression testing Alpha testing Beta testing	- Before the product is released into the market the tests that are mentioned are implemented. - By 2020 and lifetime all students must remember the concepts of these testing.

Establish relevance and make sense	
Real life context	**Tools , techniques and technologies**
reduces the level of bugs	ECUT
saves development time.	CPPTest
easier to change and refactor code.	Junit
	Jsunit
	And so on.

Engage in critical thinking	Project specification and project brief
Unit Testing It is performed under white-box testing approach. It helps the developer to decide whether the individual units of the program are working as per requirement and if they are error free or not. Avd: 1) reduces the level of bugs 2) saves development time. 3) easier to change and refactor code. **Integration Testing** Testing is done by integrating all the units together to check whether they are working without error or not necessary to check whether the software modules work in unity. System Integration testing includes a number of techniques like Incremental, Top- down, Bottom –Up, Integration techniques etc. **System Testing** **Performance testing** – • non-functional testing. • it checks the efficiency of the software. • It is done by means of load testing and stress testing where the software is put under high user and data load under various environment conditions. Avd:	All these different testing levels are being implemented in our project to detect the bugs , faults and errors and rate of failure.

- Improve user experience on sites and web apps
- Identify bottlenecks
- It tells about performance against expectations

Comparison Testing

- Also called Back-to-Back testing
- If you have multiple implementations of the same functionality then run test inputs through both implementations and compare the results for equality

Limit Testing

- Testing is done at the limits of normal use
- Test Maximum number of concurrent users ,maximum number of open file , maximum request size, maximum file size

Stress Testing

- non-functional testing.
- Test the system under extreme conditions (i.e., beyond the limits of normal use)
- The goals is to ensure that the software does not crash in conditions of insufficient computational resources.

Security testing

- non-functional testing.
- check whether the application is secured or not.
- The six basic security concepts are: confidentiality, integrity, authentication, availability, authorization and non-repudiation.

Usability testing

1) Tests the ease with which the user interfaces can be used. It tests that whether the application is user-friendly or not.
2) **black box testing** technique.
3) Five components:
   Learnability,Efficiency,Memorability,Errors,Satisfaction

Avd:

-- Better quality software
— Software is easier to use

Documentation Testing

- Test all instructions given in the documentation to ensure their completeness and accuracy
- **non-functional testing**.

Avd:

- It can start at the very beginning of the software process and hence save large amounts of money, since the earlier a **defect** is found the less it will cost to be fixed.

Regression testing

- Its purpose is to find the bugs which may get introduced accidentally because of the new changes or modification.

Advantages

- It ensures that the bugs found earlier are NOT creatable.
- can be done by automation tools
- improvs the quality of the product.

Acceptance Testing

last phase of testing where it is tested for user-interaction and response

- **Alpha testing** - It is carried out by the team of developers and they tests the system as if it is being used in work environment. Then, they try to find out how user would react to some action and how the system should respond to inputs.

- **Beta testing** –
  The product is given to the user for testing purpose.
  This is not as yet the delivered product.
  It is expected that user will bring minute problems, which were skipped to attend.
  Avd:

- Users can install, test your application, and send .

10.Coding

KEY CONCEPTS	CONCEPTS SIGNIFICANCE AND RELEVANCE
Writing the useful, extensible, and maintainable source code which can be interpreted and compiled by a computer system to perform various meaningful tasks is known as coding. 3) <u>Top down approach</u> In this approach the whole program is considered as a single entity and is fragmented into sub systems or various functions based on various characteristics. 4) <u>Bottom up approach</u> In this approach the code for the program starts from scratch considering the smallest of the components and building more complex and higher functions with the use of smaller ones. OOP tends toward Bottom-Up as we develop our objects, whereas procedural programming tends toward Top-Down as we start with one single function and gradually add to it. Structured programming is a programming technique that aims at improving the quality, development time, and clarity of a computer program by great uses of subroutines, for and while loops and block structures. Information hiding in coding is also referred to as encapsulation of the data and is done to hide information related to a particular object from view outside to that of the objects definition. Monitoring and controlling can be referred to in context of coding in a way that the coder is easily able to track down the code and fix the various bugs if any occurs in the program. The	Coding matters a lot in today's life for people pursuing this field as almost every technology that is coming up has an extensive amount of backend coding that has gone into it to create reliable systems for users benefits. Coding is like the oxygen of computer science professionals. From mobile apps to aviation technologies, all use coding for their systems. In coding things to remember as this will come in handy throughout a life time. 1) <u>The art to code:</u> a person should know how to code programs and integrate and combine them to produce high level systems for users to use. He/she should also ensure that the systems are user friendly and easily accessible. 2) <u>Information hiding:</u> also known as encapsulation is coming in high demand and usage and by 2020 will be fully utilised in almost all the applications that the humans would interact with. Its of high priority to know this phase of coding as it would be extensively used throughout and by 2050 seeing the mass number of security breaches taking place and people building much more security layers and impenetrable systems , encapsulation will be demanded a lot. 3) <u>Monitoring and control:</u> this is pretty much in use and is done by humans a lot and will come in handy till 2020 but by 2050 , robots and high end applications with much more smart applications will be used for monitoring and controlling as machines seem to be much more efficient as compared to humans. Coding is relevant to professional growth as it's the basic demand of companies to have a great coder among their team who is up to date with the latest software's and coding trends and techniques. High pay

code is maintained and crafted in such a way that it controls the program as it's the basis of it and also it monitors easy work flow of the system.

Encapsulation of the data will come in very handy for our project as we need to guard and protect the information that's being reported and not let people have unauthorized access to it. Coding will be done to a precise and appropriate level so that there isn't huge loops , or spaghetti code that is generated which may increase the output time of the whole program.

packages are also secured by people who know how to code well.

Coding is used everywhere , where technology comes to play:
1) Smartphone industry
2) Robotics
3) Networking
4) Web application based technologies
5) Graphics industry
6) Gaming technologies

REAL LIFE CONTEXT	INTERDISCPLINARY CONNECTIONS
Applications of coding in the real world can be seen in : 1) Smartphone industry Coding is used to create the operating system of the phone which is fed into the hardware of the phones system, for example android being the most widely used smart phone operating system. Its also used in app development Its used for UI/UX design and implementation. 2) Robotics Mainly machine learning and programming , use of artificial intelligence. All this requires huge sets of code written by professionals in various coding environments. 3) Networking Encapsulation of data , creating of nodes and sending packets. Describing how big the packet should be and from	Mentioned in real life context.

which node its to be transferred all this requires coding.

4) Web application based technologies
Various coding software's like php , jQuery and etc are used for the web development and for the front end coding software's like css , html are used . Programmer's code in these software's to build websites and make interactive environments for the users.

5) Graphics industry

6) Gaming technologies
Used in various game development engines such as unity, inorder to create games of high graphics . coding can involve making of gaming tools and various level generation.

7) Database implementation
Coding is used to make high end database systems for storage of various sorts of information. Mongo dB is one database engine that's used.

Coding serves all users of modern day technology. Coding is everywhere. It's in our phones, the applications we use, the laptops and pcs we carry around with us. Without coding we couldn't have achieved the world that we now live in. life would have been difficult. In today's date WhatsApp allows us to connect to peers around the globe without coding, there wouldn't have been any WhatsApp to facilitate fast connectivity with peers and we would have to wait months for our letters to be received by our peers as it was earlier. Coding has made life so much simple, it's helping in almost all industries, from hospitals, where the equipment's are fed an artificial intelligence chip that contains coded programs and functions to be performed to the fashion industry where people use software's for further enhancement of the photographs and videos.

ENGAGE IN CRITICAL THINKING	TECHNIQUES, TOOLS, TECHNOLOGY

In the world of coding , there are now a vast number of coding environments which are used for vast number of purposes.
In context of coding , it all depends on the programmers choice as to which application he wishes to build .
For example ,
In case of building android apps , the programmer would need knowledge of java programming as android is a java based system, and would need to know a to z knowledge of android suite to create great applications for the android mobiles.
 	
In case of a web developer , it gets diverged into two different parts ,
  1) Front end, that requires the developers to have mainly knowledge on html, css , JavaScript . Apart from that he should also know adobe Photoshop, illustrator and Dreamweaver to create interactive and beautiful user interfaces for the users.
    These all have different coding styles and are governed in various sorts of rules which are different from one another's coding patterns.
  2) Back end that requires php, jQuery, both of them are different and can be used for creating various functions and links for the web pages.
    So basically it all depends on the users demand and the choice of the programmer as to which coding environment he would prefer to work in and every software has its own syntax although there is a link in all of them but they are different to one another. For example syntax of python wont function in a c program and vice versa
 	
C and c++ are the most widely used programming softwares and should be known
	Coding as such doesn't need any specific technology but only the desired compilation environment to run the syntax/code efficiently. Various coding environments are :
	Html
	Css
	Java script
	J query
	Ajax
	Bootstrap

by anyone and everyone who wishes to pursue this field as it forms the basis of our coding.	

PLAN PROJECT MANAGEMENT	PROJECT SPECIFICATION AND PROJECT BREIF
NA	For our project Hybrid application for warehouse management system we need coding for both front end that is HTML, CSS, bootstrap and for backend that is java script, ajax and j query. These coding help to view the reports to the customer (manager) and also show various graphical reports. Customer can also create purchase order creation.