First: Introduction Chapters

Chapter one: Introduction

Chapter outline:

1. Software Development and Systems Analysis and Design.
2. The System Development Life Cycle (SDLC).
3. Iterative Development.

The Difference between Computer Application (App) and Information System:

Computer application: a computer software program that executes on a computing device to carry out a specific set of functions.

زي مثلا لو عندي برنامج في Two Functions واحدة بتاخد داتا عن شخص و التانية بتطبعه لما
اجي اشغل ال Two Functions دول علي الكمبيوتر و اليوزر يدخل الداتا بتاعته و تطبع ده كده
اسمه App

Information System: a set of interrelated components that collects, processes, stores, and provides as output the Information needed to complete business tasks.

زي مثلا لو انا عندي منشأة (مدرسة) و باخد داتا عنها زي المدرسين و الطلاب و الكورسات و هكذا
بتكون الداتا هما مرتبطة ببعض لانهم بيمثلوا المدرسة و بعوز اجمع الداتا دي عشان اخلي لكل
كورس مدرس او اكتر و اخلي لفصل معين مجموعة من الطلاب و هكذا و بخزن الداتا دي و بقدم
المعلومات دي كلها ك ناتج يخدم المدرسة دي

The Difference between Systems Analysis and Systems Design:

Systems Analysis: Those activities that enable a person to understand and specify what an Information system should accomplish – What that a system must do to satisfy the need or solve a problem.

الأنشطة اللي بتمكن الشخص انه يفهم السيستم ده بيعمل ايه أو المفروض يعمل ايه عشان يحل المشكلة دي

System Design: Those activities that enable a person to define and describe in detail the system that solves the need – Describe in detail How the System Will Work.

الأنشطة اللي بتمكن الشخص اني يفهم بشكل تفصيلي اكتر ازاي السيستم بيحل المشكلة دي و ازاي بيشتغل

Software Development Steps:

1. Understand the need.
2. Capture the vision.
3. Define a solution.
4. Communicate with the vision and solution.
5. Build the solution.
6. Confirm that the solution meets the need.
7. Launch the solution system.

الخطوات اللي بيمر بيها أي بناء سوفت وير هي ان افهم المشكلة اللي بيتبني عشان السوفت وير و ابني تصور أو تخيل لحل المشكلة بعدها اعرف حل للمشكلة و أتاكد انه متوافق مع التخيل اللي حطيته بعدها ابني الحل ده و أتاكد انه بيحل المشكلة بعدها أنشر السيستم بتاعي

System Development lifecycle (SDLC): The entire process consisting of all activities that required to build, launch, and maintain an Information System.

بتتكون من الأنشطة اللي المفروض أعملها عشان Software Developmentعملية داخل ال
أبني و أشغل برنامج جديد وأقدر اعمله صيانة او تحديث بعد كده

Six Core Processes of SDLC:

1. Identify the Problem (need) and obtain approval.
2. Plan and monitor the project.
3. Discover and understand the details of the problem (business task)
4. Design the System components that solve the problem
5. Build, test, and integrate the system components.
6. Complete the system tests and then deploy the solution.

خطوات ال SDLC ان اعرف المشكلة و ابدي موافقة لحل المشكلة دي بعدها اخطط للبروجيكت بعد كده
اجمع معلومات اكتر عن المشكلة و اجمع كل ال requirements بتاعه حل المشكلة دي بعد كده اصمم
مكونات النظام و أتاكد انها بطلع ال output اللي المفروض يطلع بعدها أنشر الحل بتاعي

Project: a planned undertaking that has a beginning and end and that produces definite results.

Is used to develop an Information System

Requires knowledge of systems analysis and systems design tools and techniques.

بروجيكت هو شئ مخطط له له بداية و له نهاية و بيقدم نتائج محددة يستخدم لبناء نظم معلومات و يتطلب
معرفة ادوات و تقنيات تحليل و تصميم نظم المعلومات

Information Systems Development Process (Methodology): The actual approach used to develop a particular Information system such as (Agile–Iterative–Scrum–XP–UP)

ال Methodology هو الأسلوب اللي بستخدمه عشان أنفذ ال SDLC علي سوفت وير معين

Agile Development: An Information System development process that emphasizes flexibility to anticipate new requirements during development – Fast and responsive to change.

عبارة عن Methodology بيستخدم في ال Software Development عشان هو مناسب اني احقق متطلبات جديدة في السيستم اثناء التنفيذ و سريع في اني اخلص البروجيكت في اسرع وقت و مناسب للتعديل

Iterative Development: an approach to system development in which the system is grown piece by piece throughout multiple iterations.

عبارة عن Methodology بيستخدم في ال Software Development بيتقسم فيه السيستم الي مجموعة من الاجزاء و بتعامل مع كل جزء علي حدي

Subsystem: an identifiable and fully functional part of a complete system.

جزء معرف من سيستم متكامل

Benefits of Iterative and Agile SDLC:

1. Parts of the often system can be deployed sooner.
2. Many tough problems can be addressed early in the project
3. The entire development will be flexible and able to address new requirements throughout the project.

اقدر اني اسلم اجزاء مهمه في السيستم في وقت قصير و أكتشف مشاكل قوية في السيستم في بدايته و السيستم بيكون سلس في اني أهندل متطلبات جديدة أثناء التنمية

Chapter Six: Foundations for System Design

CHAPTER OUTLINE:

- What is System Design?
- Design Activities
- System Controls and Security

Two Levels of Design

Architecture Design: Broad design of the overall system structure and also called General Design and Conceptual Design.

بيوصف تصميم من بره بدون تفاصيل كتيره و ليه اسماء تانيه

Detailed Design: Low level design that includes the design of the specific program details (Design for each use case / Design of the Database / Design of user and system interfaces / Design of controls and security)

بيوصف تفاصيل تصميم السيستم زي الداتا بيز و السكيورتري و هكذا

Design Activities

1. Describe the environment.
   ازاي السيستم هيتعامل مع السيستمس التانيه و التكنولوجيس الموجوده
2. Design the application components.
   اجزاء السيستم نفسها زي الداتا بيز و السيكيورتي و الواجهه و ازاي يتعاملوا مع بعض لما اجي أنشر السيستم بتاعي
3. Design user interface.
   ازاي اليوزر هيتعامل مع السيستم و يتواصل معاه
4. Design the Database.
   ازاي الداتا بتتخزن و ايه الداتا اللي محتاجها و العلاقات ما بينهم
5. Design the Software classes and methods.
   الكود نفسه

Design the Environment

Two Key elements Describe this term:

External Systems: needs additional information about incoming and outgoing messages.

بتحتاج المسدجات بتوصف اللي داخل و ايه اللي هيخرج و لو في اي خطأ حصل و انه اتصل بالنيتورك و هكذا

Technology Architecture: Set of computing hardware, network hardware and topology, and system software employed by an organization.

عباره عن مجموعه الاجهزه الاخري اللي بتتواصل مع السيستم الجديد اللي بيجمهم منشأه واحده

Design the Application Components

Application Component: a well – defined unit of software that performs one or more specific tasks.

جزء من السيستم بيعمل تاسك معينه

Needs to Decide محتاج تعرف

- Size / Scope حجم و حاجه الجزء ده
- Programming language يكون متبرمج ب لغه ايه
- Build or Buy هعمله انا و لا هشتريه

Define software architecture بعمل هنا البنيه بتاعه السيستم

Three layer or MVC Pattern (Architecture Design).

Detailed design of each use case بصمم اليوز كسيس

Design the user Interface

Dialog design begins with requirements (Use Case flow of activities – System Sequence Diagram).

عباره عن ديالوج بيبدأ بال Requirements

Design adds in screen layout, look and feel, navigation, user experience (UX).

التصميم ده بيظهر علي شاشه الجهاز + ان اغلب السيستمس دلوقتي بقت محتاجه انترفيسس

Designing a single user interface is now the exception, not the norm so, the term user interface should generally be considered plural rather than singular.

مش بتعامل مع الانترفيس كانه يونت واحده لازم اتعامل معاها كانها اكتر من يونت

Design the database

- Starting with the domain model class or ERD.
- Choose database structure (Relational – ODBMS framework).
- Design Architecture.
- Design database schema. الجدوال و الداتا و كده
- Design referential integrity constraints (Foreign key references)

A final key aspect of databases design is making sure the new databases are properly integrated with other existing databases.

بيبقي في احتمال كبير ان الداتا بيز الجديده اللي هتتعمل هتتفاعل مع داتا بيزس تانيه

Design the Software Classes and Methods

These models are blueprints for software methods that will be programmed, tested, and eventually deployed.

عباره عن ال Classes و ال Methods و ال Attributes اللي بعد كده بتتبرمج و بتأكد انها صح و بعدين أنشر السيستم بتاعي

System Controls and Security

- Protect the organization assets.
- Becomes Crucial in Internet and wireless.
- User Interface Controls.
- Application Controls.
- Database Controls.
- Network Controls.

Controls: are mechanisms and procedures that are built into a system to safeguard the system and the information within it.

عباره عن التركيبات و التقنيات اللي بتحمي السيستم و المعلومات اللي جواه

The difference between Integrity Controls and Security Controls

Integrity Controls: Controls that reject invalid data inputs, prevent unauthorized data outputs, and protect data and programs against accidental or malicious tampering.

عباره عن تكامل ما بين السيستم و الداتا بيز اللي بتدعمه

Security Controls: Controls that protect the assets of an organization from all threats, with a primary focus on external threats.

بتحمى السيستم من أي اخطار خارجيه

Designing Integrity Controls

- Input Controls
- Output Controls
- Redundancy, backup, and recovery
- Fraud Prevention

Objectives

- To ensure that only appropriate and correct business transactions occur.
- To ensure that the transactions are recorded and processed correctly.
- To protect and safeguard the assets of the organization.

Input Controls

Input Controls: Controls that prevent invalid data from entering the system.

Types

Value limit Controls: Check numeric data input to ensure that the value is reasonable.

بيتاكد أن القيمه معقوله زي مثلا لما بدخل اكتر من 11 رقم في رقم الموبايل

Completeness Controls: ensure that all required data values describing an object or transaction are present.

زي مثلا لو دخلت داتا ملهاش علاقه باللي في السيستم

Data Validation Controls: ensure that numeric fields that contain codes or identifiers are correct.

بيتاكد من صحه الداتا الرقميه المرتبطه بكود زي رقم الموبايل هي موجوده و لا لا

Field Combination Controls: review combinations of data inputs to ensure that the correct data are entered.

بيتاكد من صحه الفيلد اللي بيتكون من اكتر من جزء زي العنوان

Output Controls

Output Controls: controls that ensure that output arrives at the proper destination and is accurate, current, and complete.

بتتأكد ان الداتا وصلت للمكان الصح و مش متكرره و كامله

Types

- Physical access controls to printers.

- Discarded output control.

- Access controls to programs that display or print.

- Formatting and labeling of printed outputs.

- Labeling of electronic outputs

Redundancy, backup, and recovery

Take from minutes to hours and protect software and data from hardware failure. عشان تمنع تلف الهاردوير

Fraud Prevention

Fraud triangle: a model of fraud that states that opportunity, motivation, and rationalization must all exist for a fraud to occur.

موديل بيقول ان عشان يحصل اي احتيال علي السيستم لازم يكون موجود الفرصه و الحافز و الطريقه
عشان يحصل كده

Designing Security Controls

- Access Controls.
- Data Encryption.
- Digital Signature and certificate.
- Secure Transaction.

Types

- Maintain a stable, functioning operating environment for users and application system
- Product information and transactions during transmission across insecure environments such as Networks.

Access Controls

Access Control: a control that limits a user's ability to access resources, such as servers, files, Web Pages, application programs, and database tables.

بيتحكم مين اللي مسموح ليه يستخدم المصادر الحساسه في السيستم زي الداتابيز و السرفير و هكذا

Authentication: the process of identifying users who request access to sensitive resources.

العملية اللي بتعرف المستخدمين اللي طلبوا يوصلوا للمصادر الحساسه بتاعه السيستم

Multifactor Authentication: the process of using multiple authentication methods for increased reliability.

العملية اللي بتزود كفاءه تعريف المستخدمين اللي عاوزين يوصلوا للمصادر الحساسه بتاعه السيستم

Access Controls

Access Control List: a list attached or linked to a specific resource that describes users or user groups and the nature of permitted access.

قائمه فيها المستخدمين و الامكنايه بتاعتهم جوه السيستم

Authorization: the process of allowing or restricting a specific authenticated user's access to a specific resource based on an access control list.

العملية اللي بتسمح او ترفض يوزر من انه يستخدم جزء في السيستم بناءا علي الكنترول ليست

Unauthorized Users: people who aren't allowed access to any part or function of the system.

المستخدمين الغير مسموح ليهم تماما انهم يستخدموا السيستم

Registered Users: people who are authorized to access the system.

المستخدمين المسموح ليهم يستخمدموا السيستم

Privileged Users: people who have access to the source code, executable program, and database structure of the system.

المستخدمين اللي مسموح ليهم يستخدموا المصادر الحساسه في السيستم

Data Encryption

Encryption: the process of altering data so unauthorized users can't view them.

عمليه تشفير الداتا عشان مش اي يوزر يشوفها

Decryption: the process of converting encrypted data back to their original state.

برجع الداتا المتشفرة دي لحالتها الاصليه

Symmetric Key: an encryption method that uses the same key to encrypted and decrypted the data.

Asymmetric Key: an encryption method that uses different keys to encrypted and decrypted the data.

Digital Signature & Digital Certificate

Digital Signature: a technique in which a document is encrypted by using a private key to verify who wrote the document.

تكنيك بستخدمه عشان اشفر دوكيومنت معين

Secure Transactions

Secure Sockets Layer (SSL): A Standard set of methods and protocols that address authentication, authorization, privacy, and integrity.

Transport Layer Security (TLS): An Internet standard equivalent to SSL.

IP Security (IPSec): An internet standard for secure transmission of low – level network packets.

بيحمي الداتا لما بتنقل من شبكه لشبكه

Hypertext Transfer Protocol Secure (HTTPS): An Internet Standard for securely transmitting Web Pages.

بيحمي الداتا لما بنقل من بيدج لبيدج

Chapter Fourteen: System Deployment

Chapter Outline

- Testing
- Deployment Activities
- Managing Implementation, Testing, and Deployment

Implementation Activities

- Program the software
- Unit test the software
- Identify and build test cases
- Integrate and test components

Deployment Activities

- Perform system and stress tests
- Perform user acceptance tests
- Convert existing Data
- Build training materials and conduct training
- Configure and set up production environment
- Deploy the solution

Testing

Test Case: a formal description of a starting state, one or more events to which the software must respond, and the expected response or ending state.

بتكون تسيت حقيقي في ال Business Logic بتاعه السيستم بطلع ناتج متوقع بتأكد من خلاله من صحه السيستم

Test Data: a set of starting states and events used to test a module, group of modules, or entire system
The data that will be used for a test case.

الداتا اللي بستخدمها عشان أعمل Test Case

Testing: the process of examining a component, subsystem, or system to determine its operational characteristics and whether it contains any defects.

العملية التي يتم فيها فحص جزء من السيستم أو السيستم كله عشان أحدد الخصائص بتاعته و أكتشف لو في أي خطأ

Stages of testing

- **Component testing (Unit Testing):** Individual components are tested independently; Components may be *functions* or *objects* or *coherent* groupings of these entities.
  بعمل تيست لاجزاء من السيستم الجزء ظه ممكن يبقي فانكشن مثلا
- **System testing (Integration Testing):** Testing of the system *as a whole*.
  بعمل تيست للسيستم ككل
- **Acceptance Testing:** Testing with customer data to check that the system meets the customer's needs.

  بتأكد أن السيستم بيحقق ال Requirements بتاعه العميل

Why we need Verification and Validation?

- Conforms to its specification.
- Meets the requirements of the system customer.
- Involves checking and review processes and system testing.

Requirements Validation بتأكد ان السيستم بيقابل متطلبات العميل

- Check the right product is being built.
- Ensures that the software being developed will satisfy its stakeholders.
- Checks the software requirements specification against stakeholders' goals and requirements.

Requirements Verification بتأكد ان السيستم اتعمل صح

- Check that product is being built right.
- Ensures that each step followed in the process of building the software yields the right products.
- Check consistency of the software requirements specification artefacts and other software development products.

Unit Testing

Unit test: tests of an individual method, class, or component before it is integrated with other software.

تيست لجزء من السيستم قبل ما يتكامل مع باقي السيستم

Driver: a method or class used in unit testing that simulates the behavior of a class that calls and sends parameters to the unit being tested.

هو الجزء اللي بيحصله ال Unit Test

Stub: a method or class used in unit testing that receives and displays the output from the unit being tested.

هو ال return بتاع ال Driver

Unit Test Characteristics

- It is done in isolation.
- The test data and the test are done by the programmer who wrote the code.
- It is done quickly without a large requirement for other resources.

Integration Test

Integration Test: tests of the functional behavior of a group of classes, or components.

بعمل تيست للسيستم بشكل كامل

What are the expected errors?

- Interface incompatibility
- Parameter values
- Run-time exceptions
- Unexpected state interactions

Integration testing of object-oriented software is very complex because an object-oriented program consists of a set of interacting objects.

عمليه معقده لان البرنامج الواحد في أكتر من اوبيجتس بيتعاملوا مع بعض

Acceptance Testing

Usability test: a test to determine whether a method, class, subsystem, or system meets user requirements.

Many usability tests are required because they involve functional and non-functional requirements.

Performance test or Stress test: an integration and usability test that determines whether a system or subsystem can meet time-based performance criteria.

بتأكد أن السيستم بيطلع النتايج في وقت معين

Response time: the desired or maximum allowable time limit for software response to a query or update.

الوقت اللي بياخده السيستم عشان يرد علي كويري او تحديث

Throughput: the desired or minimum number of queries and transactions that must be processed per minute or hour.

العمليات اللي اقدر اعملها في السيستم في الدقيقه او الساعه الواحده

Data needed at system startup can be obtained from these sources:

- Files or databases of a system being replaced.
- Manual records.
- Files or databases from other systems in the organization.
- User feedback during normal system operation.

System Deployment

Training is needed for end users and system operators.

Why we need System Documentation?

• Provides information to developers and other technical personnel who will build, maintain, and upgrade the system.
• Modern integrated development environment provides automated tools to support all SDLC core processes.
• Requirements and design models, such as use case descriptions, class diagrams, and sequence diagrams, are developed.
• Can "reverse engineer" appropriate changes to the Models.

Why we need User Documentation?

- Provides ongoing support for end users of the system
- Describes routine operation of the system, ex: data entry, output generation, and periodic maintenance
- Software start-up and shutdown
- Keystroke, mouse, or command sequences required to perform specific functions
- Program functions required to implement specific business procedures (e.g., the steps followed to enter a new customer order)
- Common errors and ways to correct them

System Maintenance

Submitting Error Reports and Change Requests

- Standard reporting methods
- Review of requests by a project manager or change control committee
- For operational systems, extensive planning for design and implementation.

Implementing a Change

- Identify what parts of the system must be changed.
- Secure resources (such as personnel) to implement the change.
- Schedule design and implementation activities.
- Develop test criteria and a testing plan for the changed system.

Important issues to consider when planning deployment include the following:

- Incurring costs of operating both systems in parallel.
- Detecting and correcting errors in the new system.
- Potentially disrupting the company and IS operations.
- Training personnel and familiarizing customers with new procedures.

Different approaches to deployment represent different trade-offs among:

- cost
- complexity
- risk

The most commonly used deployment approaches are:

- Direct deployment
- Parallel deployment
- Phased deployment

Direct deployment or immediate cutover: a deployment method that installs a new system, quickly makes it operational, and immediately turns off any overlapping systems.
+ Increase Simplicity
- Increase Risk

Parallel deployment: a deployment method that operates the old and the new systems for an extended time period.
+ Low operational Risk
- High Cost

Phased deployment: a deployment method that installs a new system and makes it operational in a series of steps or phases
+ Reduce risk
-Increase complexity

Change and Version Control: tools and processes handle the complexity associated with testing and supporting a system through multiple versions.

Alpha version: a test version that is incomplete but ready for some level of rigorous integration or usability testing.

Beta version: test version that is stable enough to be tested by end users over an extended period of time.

Production version, release version, or production release: a system version that is formally distributed to users or made operational for long-term use.

Maintenance release: a system update that provides bug fixes and small changes to existing features.

Second: System Analysis and Deployment Chapters

Chapter Two: System Requirements

Chapter Outline

- Systems Analysis Activities
- What are Requirements?
- Stakeholders
- Information-Gathering Techniques
- Models and Modeling
- Activity Diagram

What are System Analysis activities?

- Gather Detailed Information
- Define Requirements
- Prioritize Requirements
- Develop User – Interface dialogs
- Evaluate Requirements with Users

Requirements

System Requirements: all the activities the new system must perform or support and the constraints that the new system must meet.

كل الحاجات اللي السيستم المفروض يعملها

Functional Requirements: the activities the system must perform to support the user's work.

بتمثل كل ال Functionality بتاعه السيستم الاساسيه و ال Application logic

Non – Functional Requirements: required system characteristics other than the activities it must perform or support.

الخصائص التانيه اللي بتبقي موجودة في السيستم زي ال GUI صغر المساحه استهلاك الطاقه و هكذا

FURPS and FURPS+

FURPS: an acronym that stands for functional, usability, reliability, performance, and security requirements.

FURPS+: an extension of FURPS that includes design constraints as well as implementation, system interface, physical, and supportability requirements.

Stakeholders

Stakeholders: persons who have interest in the successful implementation of the system.

الاشخاص المهتمين بالسيستم

Internal Stakeholders: persons within the organization who interact with the system or have a significant interest in its operation or success.

الاشخاص اللي بيبقوا داخل المنشأه اللي بيتعمل ليها السيستم و بيتعاملوا معاه تعامل مباشر و مستمر

External Stakeholders: persons outside the organization's control and influence who interact with the system or have a significant interest in its operation or success.

الاشخاص اللي بيبقوا بره تأثير المنشأه و خارجها و بيتعاملوا مع السيستم

Operational Stakeholders: persons who regularly interact with a system in the course of their jobs or lives

الاشخاص اللي بتتعامل مع السيستم بشكل منتظم زي مثلا مراقب الحسابات اللي بيجي كل فتره معينة
عشان يراجع الحسابات لمنشأة معينة

Executive stakeholders: persons who don't interact directly with the system but who either use information produced by the system or have a significant financial or other interest in its operation and success.

زي العملاء اللي بيبقوا عاوزين يعرفوا المواعيد مثلا بتاعه منشأه معينه

Client: a person or group that provides the funding for the system development project.

شخص او مجموعة بيمولوا السيستم

Information Gathering Techniques

- Interviewing users and other stakeholders.

بجهز اسئلة بعدين اقابل الناس و اقعد اناقش اجاوبتهم علي الاسئلة اللي سألتها بعدين اكتب الاجوبة بشكل منظم و أمهد أن ممكن يكون فيه مقابلات تاني مع حد منهم

- Distributing and collecting questionnaires.
- Reviewing inputs, outputs, and documentation.
- Observing and documenting business procedures.

Activity Diagram أعمل ال

- Researching vendor solutions.

أشوف الحلول لمشكله زي دي قبل كده و ازاي أتحلت

- Collecting active user comments and suggestions.

اخد رأي اليوزرس و العملاء و الفيدباك بتاعهم علي الموديل اللي عملته

Modeling

Model: representation or abstraction of some aspect of a system.

وصف مجرد لجزء من السيستم

Textual Models: text – based system models such as memos, reports, narratives, and lists.

Graphical Models: system models that use picture and other graphical elements to create a diagram.

Mathematical Models: system models that describes requirements numerically or as mathematical expressions.

Reasons for Modeling?

- Learning from the modeling process.
- Reducing complexity by abstraction.
- Remembering all the details.
- Communicating with other development team members.
- Communicating with a variety of users and stakeholders.
- Documenting what was done for future maintenance/enhancement

UML

Unified Modeling Language (UML): a standard set of information system model constructs and notations defined by the Object Management Group.

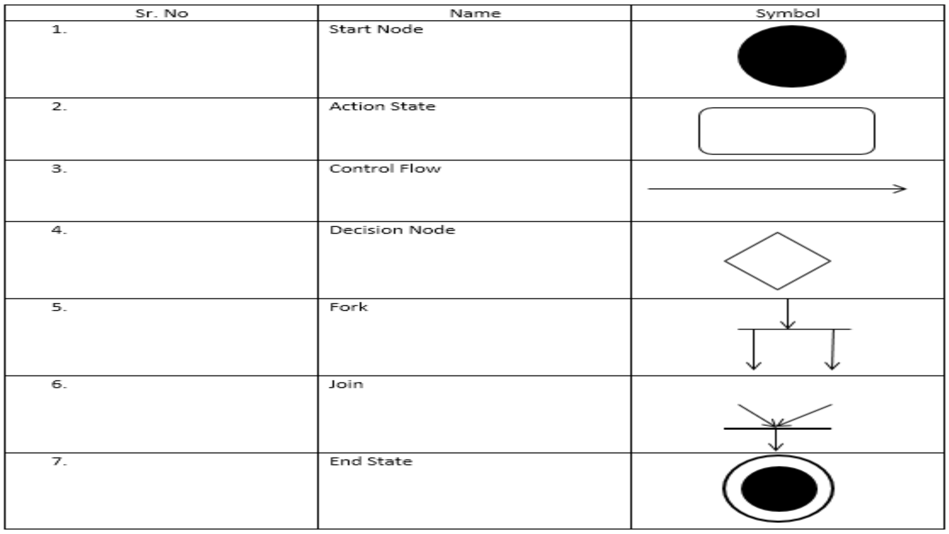
Workflow: a sequence of work steps that completely handle one business transaction or customer request.

مجموعة من الاكشنس اللي المفروض تحصل عشان أطلع ناتج معين

Activity Diagram: a UML diagram that describes user (or system) activities.

Synchronization Bar: an activity diagram component that either splits a control path into multiple concurrent paths or recombines concurrent paths.

Swimlane: an activity diagram component that divides the workflow activities into groups showing which agent performs which activity.



Sr. No	Name	Symbol
1.	Start Node	●
2.	Action State	▭
3.	Control Flow	→
4.	Decision Node	◇
5.	Fork	
6.	Join	
7.	End State	◉

Chapter Three: User Stories & Use Cases

Chapter Outline

- User Stories and Use Cases
- Use Cases and the User Goal Technique
- Use Cases and Event Decomposition

System Views

Structural View: Static system aspects (Use Case Diagram – Class Diagram).

Dynamic View: Behavior of different aspects of the system (Sequence Diagram – Activity Diagram – State Machine Diagram).

User Stories

User Story: one short sentence in the every-day language of the end user that states what a user does as part of his or her work.

Use of the User Stories

Document the functional requirements quickly and less formally than traditional requirements modeling.

جملة واحدة بتوصف اليوزر ده بيعمل ايه في السيتم و عشان ايه

Acceptance criteria: These indicate the features that must be present for the user to be satisfied with the resulting implementation.

بتوصف الحاجات اللي المفروض تتوفر لليوزر في ال Implementation

Use Cases

Use Case: an activity that the system performs in response to a request by a user and the name of each use case is Verb – Noun.

بتوصف السيستم بيعمل ايه لما اليوزر بيطلب حاجه معينه

There are two techniques for identifying use cases ***(User Goal Technique – Event Decomposition Technique)***

User Goal Technique

User Goal Technique: a technique to identify use cases by determining what specific goal of objectives must be completed by the system for the user.

بيعتمد علي معرفة أهداف اليوزر اللي لازم تتهندل بواسطه السيستم

User Goal Technique Steps

1. Identify all the potential users for the new system.
2. Classify the potential users in terms of their functional role.
3. Further classify potential users by organizational level.
4. For each type of users, interview them to find a list of specific goals they will have when using the new system.
5. Create a list of preliminary use cases organized by type of user.
6. Look for duplicates with similar use case name and resolve inconsistencies.
7. Identify where different types of users need the same use cases.
8. Review the completed list with each type of user and then with interested stakeholders.

Event Decomposition Technique

Event Decomposition Technique: a technique to identify use cases by determining the business events to which the system must respond.

Event: something that occurs at a specific time and place, can be precisely identified, and must be remembered by the system.

شئ بيحدث في وقت و زمان معين بيكون معرف مسبقا و لازم يتنهندل في السيستم

Actor: an external agent; a person, group or external system that interacts with the system by supplying or receiving data.

Elementary Business Process (EBP): is a fundamentals business process performed by one person, in one place, in response to a business event.

External Event: an event that occurs outside the system, usually by an external agent.

حاجه من بره أثرت علي السيستم

Temporal Event: an event that occurs as a result of reaching a point in time.

حاجه جوه السيستم بس بتخلص في مده معينه

State Event (Internal): an event that occurs when something happens inside the system that triggers some process.

حاجه بتحصل جوه السيستم

System controls: checks or safety procedures to protect the integrity of the system and the data.

Perfect technology assumption: the assumption that a system runs under perfect operation and technological conditions.

Why Event decomposition Technique?

- Events are broader than user goal.
- Help decompose right level of analysis.
- Uses perfect technology assumption.

Event Decomposition Technique Steps

1. Consider the external events in the system environment that require a Response from the system.
2. For each external event, identify and name the use case that the system requires.
3. Consider the temporal events that require a response from the system.
4. For each temporal event, identify and name the use case that the system requires and then establish the point of time that will trigger the use case.
5. Consider the state events that the system might respond to, particularly if it is a real-time system in which devices or internal state changes trigger use cases.
6. For each state event, identify and name the use case that the system requires and then define the state change.
7. When events and use cases are defined, check to see if they are required as part of analysis by using the perfect technology assumption. Do not include events that involve such system controls as login, logout, change password, and backup or restore the database, as these are put in as system controls.

Use Case Diagram

Brief use case description: an often one – sentence description that provides a quick overview of a use case.

بوصف ال Use Case دي بتعمل ايه

Use Case Diagram: The UML model used to illustrate use cases and their relationships to actors.

Automation Boundary: The boundary between the computerized portion of the application and the actors who operate the application.

المستطيل اللي برسم جواه ال Use Case

<<includes>> relationship: a relationship between use cases in which one use case is stereotypically included within the other use case.

Developing Use Cases Diagram Steps

1. Identify all the stakeholders and users who would benefit by having a use case diagram.
2. Determine what each stakeholder or user needs to review in a use case diagram. Typically, a use case diagram might be produced for each subsystem, for each type of user, for use cases with the «includes» relationship, and for use cases that are of interest to specific stakeholders.
3. For each potential communication need, select the use cases and actors to show and draw the use case diagram. There are many software packages that can be used to draw use case diagrams.
4. Carefully name each use case diagram and then note how and when the diagram should be used to review use cases with stakeholders and users.

Chapter Four: Domain Modeling

Chapter Outline

- "Things" in the problem Domain
- The Entity – Relationship Diagram
- The Domain Model Class Diagram
- The State Machine Diagram

Things

Problem Domain: The specific area of the user's business need that is within the scope of the new system.

مجال Business محدد اللي بيحتوي السيستم الجديد

Things: are those items users work with when accomplishing tasks that to be remembered.

هي احتياجات اليوزر اللي المفروض تكون موجودة في السيستم الجديد

There are two techniques to identify Things:

- Brainstorming Technique
- Noun Technique

Brainstorming Technique

Brainstorming Technique: a technique used to identify problem domain classes in which developers work with users to identify classes by thinking about different types of things in their work.

التكنيك المعتمد علي التعامل المباشر مع اليوزر في ايجاد الأشياء التي يجب أن تتوافر في السيستم الجديد

Steps

1. Identify a user and a set of use cases or user stories.

2. Brainstorm with the user to identify things involved when carrying out the use case—that is, things about which information should be captured by the system.

3. Use the types of things (categories) to systematically ask questions about potential things, such as the following: Are there any tangible things you store information about? Are there any locations involved? Are there roles played by people that you need to remember?

4. Continue to work with all types of users and stakeholders to expand the brainstorming list.

6. Merge the results, eliminate any duplicates, and compile an initial list.

Noun Technique

Noun Technique: a technique used to identify things in the problem domain by finding and classifying the nouns in a dialogue or description.

التكنيك المعتمد علي كتابة الأسماء التي يمكن تواجدها في السيستم الجديد

Steps

1.Using the use cases, actors, and other information about the system—including inputs and outputs—identify all nouns.

2. Using other information from existing systems, current procedures, and current reports or forms, add items or categories of information needed.

3.As this list of nouns builds, you will need to refine it. Ask these questions about each noun to help you decide whether you should include it.

4. Create a master list of all nouns identified and then note whether each one should be included, excluded, or researched further.

5. Review the list with users, stakeholders, and team members and then refine the list of things in the problem domain.

Association

Attributes: Descriptive pieces of information about things or objects.

Key: an attribute the value of which uniquely identifies an individual things or object.

Compound Attribute: an attribute that consists of multiple pieces of information.

Association: a UML term that describes a naturally occurring relationship between specific things.

Cardinality: a measure of the number of links in a particular relationship between a thing and one or more other things.

Multiplicity: a measure of the number of links in a particular association between a thing and one or more other things.

Binary Association: associations between exactly two distinct types of things.

Unary Association: association between two instances of the same type of thing.

Ternary Association: association between exactly three distinct types of things.

Nary Association: association between n distinct types of things.

ERD

Data Entities: the term used in ERD modeling to describe things about which the system needs to store information.

Entity – Relationship – Diagram (ERD): a diagram consisting of data entities, their attributes, and their relationships.

Semantic net: a graphical representation of an individual data entity and its relationship with other individual data entities.

Class Diagram and Domain Class Diagram

Class: a category or classification of a set of objects or things.

Domain Classes: classes that describes objects from the problem domain.

Class Diagram: a diagram consisting of classes and associations among the classes.

Domain model class diagram: a class diagram that only includes classes from the problem domain. مفيش فانكشنس

Association Class: an association that is also treated as a class; often required in order to capture attributes for the association.

Generalization/Specialization: A hierarchical relationship where subordinate classes are special types of the superior classes.

Superclass: The superior or more general class in a Generalization/Specialization hierarchy.

Subclass: The subordinate or more specialized class in a Generalization/Specialization hierarchy.

Inheritance: The concept that subclasses inherit characteristics of the more general superclass.

Abstract Class: a class that allows subclasses to inherit characteristics but never gets instantiated.

Concrete Class: a class that can have instances.

Whole – Part relationships: relationships between classes in which one class is a part or a component portion of another class.

Aggregation: a type of whole – part relationship in which the component parts also exist as individual objects apart from the aggregate.

اقدر اجيب جزء من السيسم و احطه في سيستم تاني و العكس في ال composition

Composition: a type of whole – part relationship in which the component parts cannot exist as individual objects apart from the total composition.

State Machine Diagram

State: a condition during an object’s life when it satisfies some criterion, performs some action, or waits for an event.

Transition: The movement of an object from one state to another state.

State Machine Diagram: a diagram showing the life of an object in states and transitions.

Pseudo state: The Starting point of a state machine diagram, Indicated by a black dot.

Destination State: for a particular transition, the state to which an object moves after the completion of a transition.

Origin State: for a particular transition, the original state of an object from which the transition occurs.

Action – expression: descriptions of the activities performed as part of a transition.

Guard – Condition: a true/false test to see whether a transition can fire.

Concurrency: the condition of being in more than one state at a time.

Path: a sequential set of connected states and transitions.

Concurrent Paths: When one or more states in a path are parallel to one or more states in another path.

State Machine Diagram Steps:

1.Review the class diagram and select classes that might require state machine diagrams

2.For each class, make a list of status conditions (states) you can identify

3.Begin building diagram fragments by identifying transitions that cause an object to leave the identified state

4.Sequence these states in the correct order and aggregate combinations into larger fragments

5.Review paths and look for independent, concurrent paths.

6.Look for additional transitions and test both directions

7.Expand each transition with appropriate message event, guard condition, and action expression

8.Review and test the state machine diagram for the class

- Make sure s state is really a state for the object in the class.
- Follow the life cycle of an object coming into existence and being deleted.
- Be sure the diagram covers all exception condition.
- Look again for concurrent paths and composite states.

Chapter Five: Use Case Modeling

Chapter Outline

- Use Case Description
- Activity Diagrams for Use Cases
- The System Sequence Diagram
- SSD Notation
- Use Cases and CRUD.
- Integrating Requirements Models.

Use Case Diagram

Use Case Description: a textual model that lists and describes the processing details for a use case.

Scenarios: a unique set of internal activities within a use case.

Preconditions: conditions that must be true before a use case begins.

Postconditions: What must be true upon the successful completion of use case.

SSD

System Sequence Diagram (SSD): a diagram showing the sequence of messages between an actor and the automated part of the system during a use case or scenario.

SSD Steps

1. Identify Input message.
2. Describe the message from the external actor to the system using the message notation.
3. Identify any special conditions on input messages.
4. Identify and add output return values.

Lifeline: the vertical line under an object on a sequence diagram to show the passage of time for the object.

Loop frame: notation on a sequence diagram showing repeating messages.

True/false condition: part of a message between objects that is evaluated prior to transmission to determine whether the message can be sent.

opt frame: notation on a sequence diagram showing optional messages.

Alt Frame: notation on a sequence diagram showing if-then-else logic.

CRUD Technique

CRUD technique: an acronym for create, Read/Report, Update, and Delete; a technique to validate or refine use cases.

CRUD Steps

1.Identify all domain classes

2.For each class verify that use cases exist to

- Create a new instance
- Update existing instances
- Reads or reports on information in the class
- Deletes or archives inactive instances

3.Add new use cases as required.

4.Identify responsible stakeholders

5.Identify which application has responsibility for each action: which to create, which to update, which to use.

Chapter Ten: Approaches to System Development

Chapter Outline

- *The System Development Life Cycle.*
- *Methodologies, Models, Tools, and Techniques.*
- *Agile Development.*
- *The Unified Process, Extreme Programming, and Scrum.*

Three important trends affecting system development

- *Consumer Device.*
- *Distribution of application software.*
- *Continuing movement toward Web – based applications.*

Plan-driven and agile processes

Plan – Driven: processes of processes where all of the process activities are planed in advance and progress is measured against this plan.

عباره عن خطه للمشروع فيها Activities معينه بقعد أقارن فيها انا نفذت ايه من المطلوب في وقت قد ايه

Agile Processes: planning is incremental and it is easier to change the process to reflect changing customer requirements.

التخطيط فيها بيكون تدريجي و Flexible اوي في التغير

Most practical processes include elements of **both** plan-driven and agile approaches.

عاده ال Plan – Driven و ال Agile Processes بيشتغلوا مع بعض مفيش واحده منهم بتيجي لوحدها

There are no right or wrong software processes.

مفيش حاجه اسمها ان في Methodology وحشه كلهم بيادوا نفس الغرض

Extreme Programming (XP)

XP: Uses an iterative SDLC and based on Agile principles.

One of the original agile development methodologies from Kent Beck.

Extreme: often thought to be radical, but really just focuses intently on industry best practices and combines best practices in new ways.

معناها اني اركز علي صناعه السوفت وير نفسه و اعملها بطرق جديده

XP is based on **core values**

- Communication
- Simplicity
- Feedback
- Courage

XP also defines a set of **XP practices**

XP Core Values

Communication: one of the major causes of project failure is a lack of open communication among the right players at the right time and at the right level.

يكون في تواصل ما بين الشخص الصح في الوقت الصح في المكان الصح و أكون متأكد من ان اللي بعمله ده متوافق مع معايير المنشأه دي

Simplicity: XP includes techniques to reinforce keeping things simple to make it a standard way of developing systems.

بخلي كل حاجه ابسط ما يمكن عشان يبقي في طريقه ثابته عشان أبني السيستم

Feedback: as with simplicity, getting frequent, meaningful feedback is recognized as a best practice of software development.

باخد رأي العملاء في اللي انا بعمله

Courage: developers always need courage to face the harsh choice of doing things right or throwing away bad code and starting over.

المطور لازم يكون عنده الشجاعه انه يبدأ كود جديد خالص عن تعديل كود موجود بس في أخطاء كتيره

XP Practices

Planning: XP planning focuses on making a rough plan quickly and then refining it as things become clearer. This reflects the Agile development philosophical dictum that change is more important than detailed plans.

بتركز علي انك تعمل خطه قويه و سريعه عشان كل حاجه تبقي واضحه

Testing: XP intensifies testing by requiring that the tests for each use case (story) be written first before the solution is programmed.

بعمل تيست بكل يوز كييس أو يوزر استوري قبل ما اعملها الكود بتاعها

Pair Programming: XP practice in which two programmers work together on designing, coding, and testing software.

أن كل اتنين شغالين مع بعض بيعملوا الديزاين و الكود و التيست و كل واحد بيعالج اخطاء التاني

Simple Design: XP conforms to the principles of Agile Modeling.

It accomplishes the desired result with as few classes and methods as possible and that doesn't duplicate code.

ال XP بيأكد علي مفاهيم ال Agile Modeling عشان يحقق النتائج المطلوبة عشان كده لازم الديزاين يكون Simple و اخلي عدد ال Classes و ال Methods أقل ما يمكن

Refactoring the Code: refactoring is the technique of improving the code without changing what it does. XP programmers continually refactor their code to achieve a simpler design.

تكنيك بستخدمه عشان أزود كفاءه الكود بدون ما أغير فيه عشان اخلي الديزاين أبسط ما يمكن

Owning the Code Collectively: In X, everyone is responsible for the code. Collective ownership allows anyone to modify any piece of code.

كل واحد في التيم مسئول عن البروجيكت طالما كل واحد يقدر يعدل في أي جزء من الكود

Continuous Integration: This practice embodies XP's idea of "growing" the software. Small pieces of code which have passed the unit tests are integrated into the system daily or even more often.

الكود بيتقسم الي اكتر من جزء لما اجي ابني كل جزء لازم اعمله تيست و أشوف هل في تكامل معاه و مع السيسم كله و الاجزاء اللي اتعملت قبل كده

On-Site Customer: as with all adaptive approaches, XP projects require continual involvement of users who can make business decisions about functionality and scope.

لازم يكون في حد يساعدني في اتخاذ ال Business Decisions

System Metaphor: a system metaphor should be easily understood and well known to the members of the development team. It can guide members toward a vision and help them understand the system.

لازم السيستم يكون مفهوم و ميكونش في كيانات معقده

Small Releases: consistent with the entire philosophy of growing the software small and frequent releases provide upgraded solutions to the users and keep them involved in the project.

مستناش ان السيستم يخلص بالكامل و بعدين اديه للعميل لكن اخلص اجزاء صغيره منه و اوريها للعميل لحد ما يخلص كله

Forty Hour Week: The exact number of hours a developer works isn't the issue. The issue is that the project shouldn't be a death march that burns out every member of the team.

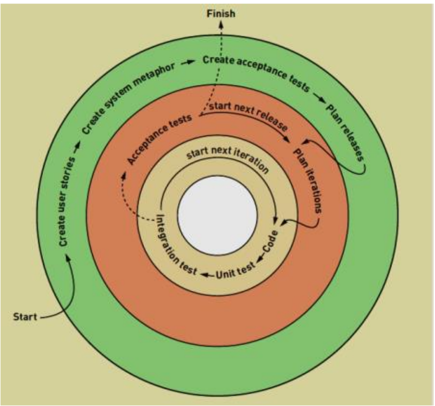
مش لازم اشتغل 24 ساعه لكن ارتب نفسي بحيث أن مطلعش منتج ردئ بسبب عدم التنظيم

Coding Standards: developers should follow standards for coding and documentation.

لازم يكون في حاجه ثابته يمشي عليها كل المبرمجين عشان يبنوا السيستم

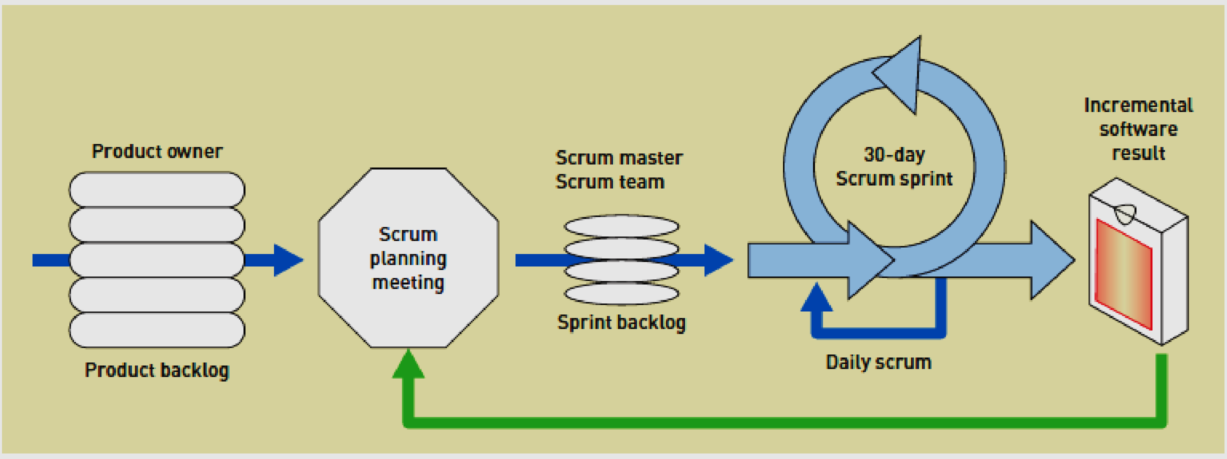
XP Activities

- Project Activities
- Release Activities
- Iteration Activities



Scrum

- *Another influential agile, iterative development methodology based on ideas from Rugby.*
- *Scrum philosophy is the complete control a team exerts over its own organization and its work processes.*
- *Software is developed incrementally, and controls are imposed empirically by focusing on things that can be accomplished.*



Scrum Organization

Product backlog: a prioritized list of user requirements used to choose work to be done in a Scrum project. Only a few of the high-priority items are worked on at a time.

بيكون فيه ال Requirements و ال priorities

Product owner: *The client stakeholder for whom the system is being built Responsible for project backlog and priorities.*

اللي بيتعمل السيستم عشانه و هو اللي بيحدد ال Requirements و ال priorities

Scrum master: *The person in charge of a Scrum project—similar to a project manager.*

الليدر بتاع التيم

Scrum team: It is usually 5 to 9people. Scrum team sets own goals, organizes self, makes decisions.

بيكون مسئول عن الاهداف و تنظيم البروجيكت و اتخاذ القرارات

Scrum Practices

Sprint: A time controlled mini project that implements a specific portion of a system. Firm 30days time box with specific goal or deliverable. The scope of that sprint is then frozen, and no one can change it neither the product owner nor any other users. Sprint backlog defines the scope.

بيكون ميني بروجيكت بيهندل تاسك معينه

Daily Scrum: a daily meeting of all members of the team to report progress (15 minutes max).

ميتينج يومي اقصي مده ليه ربع ساعه كل عضو في التيم بيعرف هو عمل ايه

Sprint final half day review meeting: scheduled to review and identify changes needed for the following sprints.

بعرف فيه هنعمل ايه في ال Sprints الجايه

Third: Design Chapters

Chapter Seven: Defining the System Architecture

CHAPTER OUTLINE

- Anatomy of a Modern Information System.
- Architecture Concepts.
- Interoperability
- Architecture Diagrams.
- Describing the Environment.
- Designing Application Components.

Server: a computer or group of computers that manages shared resources and enable users and other computers to access those resources via network.

بيكون عباره عن جهاز كمبيوتر كبير بيشارك Resources معينه لاجهزه تانيه من خلال نيتورك

Application Software: software that performs user or business tasks. It may be an app or web – based application.

برنامج بيأدي غرض معين و ممكن يكون موقع الكتروني

App: application that is installed on storage devices of a computer or cell phone.

البرنامج اللي بسطبه علي اجهزه تخزين الكمبيوتر او علي الموبايل

Web – based application: application software that uses a web browser as the user interface, has a URL for application access and uses the web servers.

بيكون ابليكشن بيستخدم ال browser ك interface و بيكون ليه URL يقدر من خلاله اليوزر يستخدموا الموقع ده

System software: software such as OS and servers behind the scenes to support applications.

بتكون ابليكيشنس بتخدم ال Applications التانيه زي نظام التشغيل و السيرفرات

Embedded Software: applications implemented in other applications to perform additional features such as graphical interface, voice recognition, speech generation and so on.

ابليكشنس جوه ابليكشنس تانيه بتعمل خدمه زياده زي انترفيس معينه – اليسرش عن طريق التحدث شبه ال APIs

The types of the embedded software

- **Toolbars:** A set of links to Web resources or installed apps that extend the capabilities of a Web browser.
- **Widgets:** An example of a Web browser widget is a time-and-temperature or stock market widget that is always displayed in one corner of the browser no matter what Web page is being viewed.
- **Plug – ins:** Web browser plug-ins are often used to correctly display certain types of Web content.

Protocol: a set of languages, rules, and procedures that ensure accurate and efficient data exchange and coordination among hardware and software components.

مجموعه من الخطوات اللي بتم عشان تعمل تاسك معينه

It has two types

- Network Protocol

  Virtual private network: secure communication over the internet using technologies that reliably identify senders and recipients and encrypt network messages sent among them.

  طريقه تواصل امنه خلال الانترنت بتعرف الراسل و المستقبل و بتشفر المسدجات ما بينهم

- Web Protocol

  HTML: a protocol that defines the structure and content of a web page.

  برتوكول بعرف بيه ال Content بتاع اي ويب بيدج

  XML: an HTML extension that enables the meaning of word, phrases, or numbers to be defined.

  شبه ال HTML بس بيعرف ال word و ال numbers ك Tags

  HTTP: a protocol that defines the format and content of requests for web documents and related data communication.

  برتوكول بستخدمه عشان get – post داتا من – الي ويب بيدج

Software as a service (SaaS): a software delivery model similar to a utility. In which application software is accessed via the internet without locally installed programs.

اللي هو ال API

Web service: software function or related set of functions that can be executed via web standards.

برامج معينه بستخدمها بس في الويب بيدجس

Client/server architecture: a software design and deployment method that divides software into components that manage resources and components that use those resources.

There layer architecture divides an application into three layers

View layer: Contains the user interface.

بيمثل الانترفيس سواء window form او web browser

Business logic layer: Contains the programs that implement the business rules and processes.

بيمثل ال Controller في ال MVC

Data layer: Interacts with the data.

الداتابيز نفسها

Interoperability: the ability of a component or system to interact with other components or systems.

امكانيه تفاعل جزء من السيستم مع باقي الاجزاء

Network diagram: a model that show how locations and hardware components are inter-connected with network devices and writing.

موديل بيوضح ازاي ال Components بيتواصل مع النيتورك

System of record: a system or application component that maintains the current and correct master copy of one or more data items.

Top-down approach: Considers entire system as a single component performing all of the functions described during analysis activities.

بيتعامل مع ال Single Component انه بيعمل كل ال functions بتاعه السيستم كلها

Bottom-up approach: Considers each function separately and looks for similarities as a basis for grouping software that implements the functions into larger application components.

بيتعامل مع ال Single Component انه بيعمل function في السيستم

Chapter Eight: User Interface

CHAPTER OUTLINE

- Understanding the User Experience and the User interface.
- Fundamental Principles of User – Interface Design.
- Transitioning from Analysis to User – Interface Design.
- User – Interface Design.
- Designing Reports, Statements, and Turnaround Documents.

The difference between UX & UI

User experience (UX): all aspects of a person's interaction with a software application, including actions, responses, perceptions, and feelings.

يعني انطباع المستخدم في واجهه البرنامج

User interface (UI): a set of inputs and outputs that the user interacts with to invoke the functions of an application.

مجموعة من المدخلات و المخرجات اللي المستخدم بيتفاعل معاها عشان يستخدم مزايا البرنامج

User Centered Design

User – centered design: design techniques that embody the view that the interface appears to be the entire system.

التكينكات المستخدمة الرئسية اللي بتجسد شكل البرنامج عشان المستخدم يتفاعل معاها

Usability: the degree to which a system is easy to learn and use.

يعني اني اقدر احقق اني المستخدم يلاقي سهولة في تعلم البرنامج و استخدامه

Human – computer interaction (HCI): a field of study concerned with the efficiency and effectiveness of user interface vis -a – vis computer systems.

المجال اللي بيهتم بدراسة تفاعل الانسان مع الالة

Three Principles of User – Centered Design

- Focus early and throughout the project on the users and their work.
- Evaluate all designs to ensure usability.
- Use iterative development.

Metaphors: analogies between features of the user interface and aspects of physical reality with which users are familiar.

تمثيلات (ايماءات) ما بين مميزات الانترفيس و مكونات السيستم الفعلية زي مثلا icon ال email
بدل علي ان المكان ده بيتكتب فيه mail و كذلك زي ال phone و هكذا

Direct manipulation metaphor: metaphor in which objects on a display are manipulated to look like physical objects or graphic symbols that represent them.

Desktop metaphor: metaphor in which the visual display is organized into distinct regions.

Document metaphor: metaphor in which data is visually represented as paper pages or forms.

Dialogue metaphor: metaphor in which user and computer accomplish a task by engaging in a conversation or dialogue via text, Voice, or tools, such as labeled buttons.

User Interface Design

Human – interface objects (HIOs): icons and other objects on a screen that can be manipulated by the user and cause some action to occur.

Affordance: when the appearance of a specific control suggests its functions.

يعني لما مظهر اي كنترول في الانترفيس يحقق الميزه بتاعته

Tool tip: brief instructions that pop up when the mouse hovers over a control.

التيكست اللي بيظهر علي ال icon لما اخلي الموس تقف عليه

Visibility: when a control is visible so that users know it is available.

مفهوم اني اليوزر يعرف ان الحاجه دي موجوده لما تظهر قدامه في الانترفيس

Feedback: some visual or audio response by the system in response to a user action.

رد سماعي او كتابي بواسطة السيستم للرد علي اكشن معين من اليوزر

Radio buttons: a group of selection items that allows only one item to be selected within the group.

يعني عندي مجموعة items كتيره المفروض اختار واحده منها بس

Check boxes: a group of selection items that allows either none or many items to be selected within the group.

يعني بيكون عندي مجموعة items كتيرة و المفروض اني ممكن مختارش و لا واحدة او اني اختار كتير

Continuity: maintaining a certain level of consistency over time, across multiple releases.

يعني احافظ بمرور الوقت علي ثبات الانترفيس

Discoverability: a feature of the user interface that provides clues to help the users uncover hidden features.

Active discovery: a user – interface feature to lead users to discover hidden features.

الميزة اللي بتعرف ال user لو في اي features مش ظاهرة قدامه

Breadcrumbs (navigation): the technique of displaying the sequence of pages traversed to allow easy backing out to a previous page.

تكنيك بيعرض كل الصفحات و بيمكنك انك ترجع للصفحه السابقة

Storyboarding: Sequences of sketches of the displaying screen during a dialogue.

مجموعة من الرسومات بتعرض كل screen فيها features ايه و بتمكنك انك تفهم ال interface

Text box: a box that accepts text from a keyboard or speech input.

يعني input بيقبل data بدخلها من خلال الكيبورد او المايك

List box: a text box with a list of predefined data values.

Combo box: a combination text box and list box that displays a predefined list, but also allows entry data.

Mobile responsive: designing Web sites so that the pages are responsive to being displayed on small, mobile devices.

Action bar: a common navigation bar with action control used for smartphones and usually placed at the bottom of the screen.

Drill down: user – interface design technique that enables a user to select summary information and view supporting detail.

Consistency

Types

- Consistency Within and Across Platforms
  Platform: Tools to build app.
  Across Platform: Windows – Android – Apple - Unix
- Consistency Within a Suite of Applications
  ما بين ال apps اللي من نفس الشركه زي MICROSOFT
- Consistency Within an Application
  ان يكون تصميم ال Items اللي في ال App واحد
- Consistency Versus Continuity
  Continuity: maintaining a certain level of consistency over time, across multiple releases.

ان احافظ علي ال Consistency لما انشر releases جديده من نفس ال Product زي اصدرات ال Windows كده.

Detailed report: reports that contain specific information on business transactions.

Summary report: reports that summarize detail of recap periodic activity.

Exception report: reports that provide details or summary information about transactions or operating results that fall outside a predefined normal range of values.

Executive report: reports used by high – level managers to access overall organizational health and performance.

Electronic Report: Electronic reports allow flexibility in the organization and presentation of information.

- Must be dynamic.
- Generate the different views as needed
- Contains (detailed and summary sections, data and graphics together, boldface type and highlighting, dynamically change their organization and summaries, hot links to related information).

Chapter Twelve: OO Design Fundamentals

Chapter Outline

- OO Design.
- Steps of OO Design.
- Design Classes and the Design Class Diagram.
- Designing with CRC Cards.
- Fundamental Principles for Good Design.

Intro to OO Design المرحلة الاخيرة من الديزاين

OO Design: **1 -** Process by which a set of detailed Object – Oriented Design models are built**,** which are then the programmers to write and test programs for the new system.

2 – The process to identify the set of classes, their methods, and their messages required to execute a use case "is use – case Driven".

- Strength of OO is requirements models from are extended to design models.
- Agile approach says create models only if they are necessary.

الديزاين اللي بيحتاجها أو بيسهل لل Developers أنهم يعملوا Write و

لكود السيستم الجديد Test و بعمل ال Models دي لما بحتاجها فقط.

Instantiation: creation of an object based on the template provided by the class definition.

هو تعريف Object اللي بيوصف سلوك معين بناءا علي ال Class المتعرف منه

Object – Oriented Event – Driven Program Flow 3 Layers

- Analysis Models. (Use Cases – SSD – Activity Diagram)
- Design Models. (Communication Diagrams – Sequence Diagrams – CRC Cards).
- Programming Models. (OO Program classes with methods).

OO Design Steps

A Single use case is chosen, and the appropriate models are constructed or updated to describe the details of the use case.

بختار Use Case معينه لو هي Simple ه Construct CRC Card لو Moderately Complex ه Construct Communication Diagram و لو هي Complex ه Construct Sequence Diagram

Stereotype: a way of categorizing a model element by its characteristics, indicated by guillemots (<< >>).

بعرف فيه الموديل ده بيمثل ايه (Entity – Boundary – Data Access - Control)

Entity Class: a design identifier for a problem domain class.

بيمثل ال Model في ال MVC

Persistent Class: an entity class whose objects must continue to exist after a system is shut down.

ال Data لازم تكون ثابته سواء في فايل او داتابيز

Boundary Class (View Class): a class that exists on a system's automation boundary, such as an input window form or web page.

بتمثل ال View في ال MVC

Control Class: A Class that mediates between boundary classes and entity classes, acting as a switchboard between the view layer and domain layer.

بتمثل ال Controller في ال MVC

Data Access Class: A class that is used to retrieve data from and send data to a database.

بستخدمه عشان اخد داتا من الداتا بيز أو ابعت داتا للداتا بيز

Design Class

Visibility: a notation that indicates (by plus or minus sign) whether an attribute can be accessed by another object.

- Attribute Name by Lower Case camelback notation.

Method Signature: a notation that shows all the information needed to invoke or call the method.

- Initial Value if applicable, the default value.
- Property if applicable, such as {Key}.
- Return Type the type of the data returned.
- EX: - accountNo: String {Key} \ startingJobCode: Integer = 01
- EX: + setName(fName , lName): void

Class level Method: a method that is associated with a class instead of with objects of the class.

Class level Attribute: an attribute that contains the same value for all objects in the system.

First Cut Design Class Diagram

- Proceed use case by use case, adding to the diagram.
- Pick the domain classes that are involved in the use case.
- Add a controller class to be in charge of the use case.
- Determine the initial navigation visibility requirements using the guidelines and add to diagram.
- Elaborate the attributes of each class with visibility and type.

Navigation visibility: a design principle in which one object has a reference to another object and thus can interact with it.

CRC Cards

CRC Cards: a brainstorming and design technique for designing interactions in use cases by assigning responsibilities (الحاجات اللي المفروض يعملها) and collaborations (هيتواصل مع مين في السيستم) for classes.

- One Card per class.
- Front has responsibilities and collaborations.
- Back has attributes needed.

CRC Cards Procedures from Book (Phase 2)

Design Principles

Object Responsibility: a design principle in which objects are responsible for carrying out system processing (Knowing (Collaborators) + Doing (Responsibilities)).

كل اوبيجت مسئول عن حاجه

Separation of Responsibilities: a design principle that recommends segregating classes into sperate packages or groupings based on a primary focus of processing responsibilities (MVC Pattern (View + Model + Controller)).

بقسم السيستم لاكتر من جزء و كل جزء مسئول عن حاجه

Protection from variations: a design principle in which parts of a system that are unlikely to change are segregated from those that will.

تغيير اي جزء في السيسم ميأثرش علي جزء تاني

Indirection: a design principle in which an intermediate class is placed between two classes to decouple them but still link them.

زي ال Adapter pattern كده

Coupling: a qualitative measure of how closely the classes in a design are linked.

مقياس بيعرفني قد ايه الكلاسيس دي مترتبطه ببعض

Cohesion: a qualitive measure of the focus or unity of purpose within a single class.

مقياس بيعرفني اليونت ده محطوط في الكلاس المناسب ليه و لا لا

Chapter thirteen: OOS Design “Use Case Realization”

Chapter Outline

- OO Design with interaction Diagrams.
- Use Case Realization with communication Diagrams.
- Use Case Realization with sequence diagrams.
- Developing Multilayer Design.
- Updating and packaging the Design Classes.
- Design Patterns.

Use Case Realization: The process of elaborating the detailed design with interaction diagrams for a particular use case.

العمليه اللي بعمل بيها ديزاين لل Complex use cases

Two Types of interaction diagrams

Communication Diagram: Type of interaction diagram that emphasizes the objects that send and receive messages for a specific use case.

Sequence Diagram: Type of interaction diagram that emphasizes the sequence of messages sent between objects for a specific use case.

Communication Diagrams Items

- Actor.
- Object.
- Link.
- Message.

Use Case Realization with Sequence Diagrams guidelines

- Determine the objectives.
- Determine what Information is needed.
- What class needs it.
- What class provides it.
- Identify the complete set of classes that will be affected by the message.
- Any classes that are listed in either the preconditions or postconditions of a use case should be included in the design.
- Other classes to include are those that are created, classes updated during the use case, and those that provide information used in the use case.

Activation lifeline: a representation of the period during which a method of an object is alive and executing.

Perfect Technology Assumption: First encountered for use cases. We don't include messages such as the user having to log on.

Perfect Memory Assumption: We have assumed that the necessary objects were in memory and available for the use case.

Perfect Solution Assumption: The First – Cut sequence diagram assumes no exception conditions.

Dependency Relationship: a relationship between packages, classes, or use cases in which a change in the independent item requires a change in the dependent item.

Design Patterns: standard design techniques and templates that are widely recognized as good practice (Controller – Adapter – Factory - Singleton).

- They are written up in design pattern catalogs / references include:
  - Pattern Name.
  - Problem that requires solution.
  - The pattern that solves the problem.
  - An example of the pattern.
  - Benefits and consequences of the pattern.