SE

Week: 2

Data: 21-03-2022

- **SDLC**

  Software Development Life Cycle

1- Requirenments

Users Stories (Why, How, what)
wireframe (User requirenments of screen
Story Board (Description)
Functional / Non-Functional / User / Busnice
Physical Product (Requirenments)

2- Analysis

3- Design

High Level
Low Level

4- Building

Coding the code in Programming Language acording to user Requirnments

5- Testing

Alpha (Testa by Developer)
Beta (Test by Users / Otherise Useirs)

6- Maintainence

1- Requirement

Formal Defination

A condition or capability that must be met by a system or its component to satisfy a content, standard, specification or other formal document.

SRS → Software Requirement Specification

Informal Defination

A specific description of client needs which can be used to create a real word application / product.

→ Needs:

→ Wants:

Use case: How user will interact with software is called user care.

Date: 1-4-2022

Types of Requirements

1. Functional

- The main requirement
- without this the software does not exist
- Behaviour that the developed product should do or support

- shall → word for functional req.
- must → word for non-functional req.

• Business Requirement

- Purpose of the project
- Business requirements are not business rules.

Example of Business rules:
1. Legal regulation
2. Privacy Policy

• User Requirement

- what the product do for the user:
- accomplish with the product

2. Non-Functional

- Performance of software - Respone time, Utilization
- Scability - more users and fonctions can be add
- Security - user must be safe from hacking
- Reliability
- Availability - 24/7 by ensuring

There are two methods to write fonctions

• Exact Method

As a user, I must log into the system with 3 seconds.

• Range Method

→ The responce time of the software must be 3 to 10 seconds

Date: 18-4-2022

Stakeholders of a Software

Anyone affected by or who has an effect on the success of the project

- End User
- Client
- Managers
- Administrator

Types

(i) **Primary**: End User who actually use the product
(ii) **Secondary**: who occasionally use the product
(iii) **Tertiary**: who are affected by the product. Make decission about the product. e.g (Developer)

User Interface:

- User Interface must be responsive
- e.g. Screen, button, text. etc.

Challenges

(i) User inability to explain what the need
(ii) Users are biased by previous bad design
(iii) Developer have trouble seeing through users point of view.

Solution

(i) Always design for intermediate level users.

User Interface / Limitations

(i) Perceptual / Sensory Limitation (Colour Blindness)
(ii) Physical Limitation (Right / Left handed)
(iii) Cognitive / Memory Limitation (Memorize Everything)
(iv) Cultural Limitation (Biased due to cultural limitation)

How to gather Requirements?

1. Interview (Client)
2. Focus Group (A page is moved in a group)
3. Observing (How every one is working / Using)
4. Consulting previous products
5. Questionnaire (Use google forms)
6. Use case
7. Prototyping (It is not actual Product but it is made before the actual product)

Date: 11-4-2022

Uses case

- To achieve paticular environment / goal.

Use case format

- Name
- Participating actor – User
- Goals – A goal of use case is what it want to achieve.
- Triggers – when a user want to buy product they will select payment method.
  Buy now – Function of Implimentation.
  Button
- **Pre-Condition** – all those conditions which must be true before starting. Sing-In, Internet etc.
- **Post-Condition** – The condition which must be true after the execution of basic flow.
- Basic flow – Alway ~~wrig~~ write in Points.
  – The flow of execution.
  – e.g Main ~~Soerio~~ Scenrio, Basic Scenrio

- Alternate Flow - Different ways to sign-In.
- Exceptions - The problem with solutions. (more then one)
  - Right down exception in points.
  - Important in testing the software.
- Testes use test cases. (Pass or Fail)
- Qualities - Performance e.g. The software must take 3 seconds to start.
  - Non-functional Requirement

View DMC

- Use Case

Name:	DMC
P. Actor:	Student
Goals:	Check GPA, CGPA
Triggers:	Request to view DMC
Pre-Conditions:	LMS account, Internet connection, Browser
Post-Conditions:	View DMC and view GPA, CGPA Successfully
Basic Flow:	1) Open browser 2) Log in account 3) Request to view DMC 4) View DMC
Alternate Flows:	No alternate flow / Not available
Exceptions:	1) Student don't have LMS Account 2) No Internet Connection 3) Don't know about DMC

Qualities:

1) User can check CGPA
2) Takes less than 3 seconds
3) Must be secured, Use password for security
4) Easy to use - must be descriptive

Wireframe

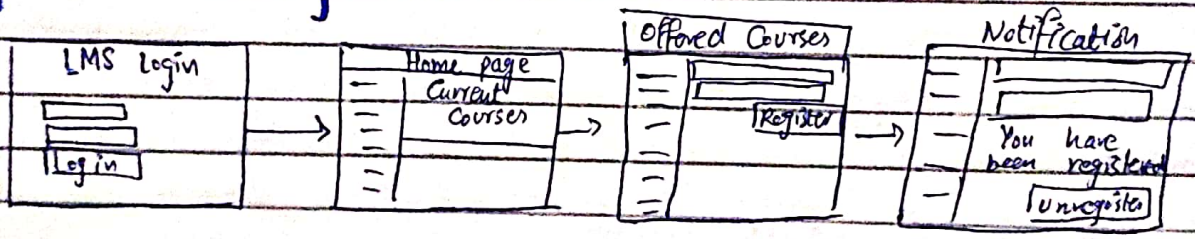
A wireframe, also known as a mock-up, can be thought of as a kind of early blueprint.
Mock-up - How the system will look like.

Proposal

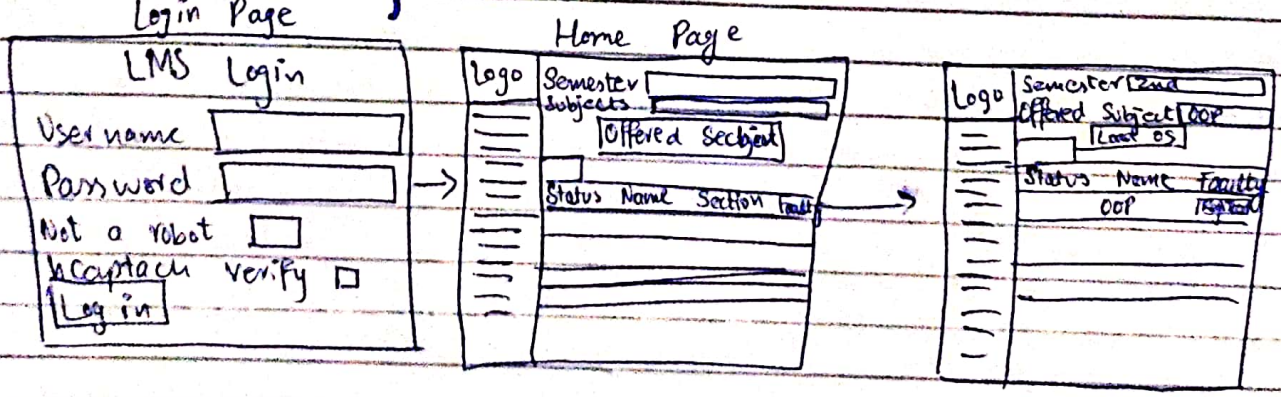
2- lines in Abstract, 1- Paragraph in Introduction
Background - 2 lines intro
Problem
Recent Solution
Your Solution
Advantages
Future Scope

Date : 15-4-2022

High Level Story Board



Low Level Story Board



Stackholder

A person who is effect positively or negatively from the software.

- User
- Owner

Epic User Story

"who", "what" and "why"

As a user, I want to register in a ~~Subject~~ Course, so I can study the Subject.

- As a user, I will Log in to Uet LMS.
- As a user, I open LMS and select current Semester. ~~and~~ ~~Subject that I want to study.~~
- As a user, I will select Loard Offered Subjects. by clicking "Loard Offered Courses" button.
- As a user, I will select the subject that I wish to study.
- As a user, I will register the ~~a~~ subject by clicking "Register" button, so I can register in subject.

Software House Methodology

Agile model —implement→ Scum

SRS - Software Requirement Specification

Date : 18-4-2022

SDLC

- **Water fall model**
  - The very first, basic model

1) Requirement Alalysis 2) Design 3) Implementation
4) Testing 5) Deployment 6) Maintenance

- Impliment all steps in sequence, step by step
- water fall model – Diagram
- Advantages } for every model
- Disadvantages }
- If there is a problem in SRS, then the whole water fall model fails.

- **Iterative model**
  - Itrerative model generates the product after every ~~Itratration~~ Iteration.
- **Big Bang model**

- **Spiral model**
  - The only model that allow to risk analysis
  - POC – Proof of Concept
- **V-model**
  - Verification model – Check the requirement seperately
  - Used to resolve maximum errors in software
- Testing a software according to user wants is called Acceptance testing.
- **RAD model**
- rapid application development model

Date: 25-4-2022

Agile - Scrum

- Agile approaches - Scrum - extreme Programming

Artifacts: Sprint Bachrote rog

- list of work that a develop team must address