

golden ratio

Why: Purpose

How: Process

What: The result

Demoz.com / Alibaba

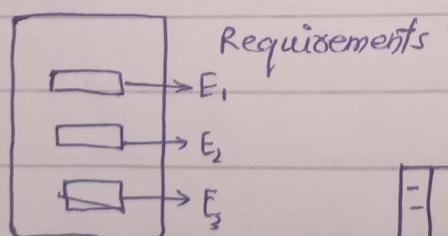
why: Shop from everywhere

How: System must be easy
to use

SDLC | (why frame)
wire

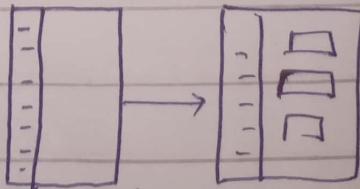
Requirements stories
usage (Storage) (why, How, what)

wireframe
Story board / board



LMS example

Registered subjects
View MC



Story board: Graphical representations of
user requirements

1 2 3 4 5
Fun | non fun | user | Business Req | Physical Products

1: main function of the system

2: no performance, security, no response time

It's important as function requirements.

Performance and efficiency tells by non-fun

3: Each and every screen (board. Story)

4: Profit, time save

5: like a product to run in water, it

works efficiently but not water proof)

↳ physical requirements

Analysis

what are you dealing to process.

GPU, image processing for camera to

select where a person is wearing helmet/

Design → high level → high level view of system
↳ low level

Link CS employee → fee → registered subject
Exam - Employee → fee → exam

Building / Development

Just select the lang and build your code.

Testing

we donot deliver software /product until we don't test it.

- alpha - Y: design, singn up , F: design 2nd page, check /test ^{others} _{team} page.

- Beta

like game and errors comes and feedback to improve it.

Check by non-development stage not included persons.

→ Unit testing (Part by Part)

→ System testing (as whole)

Maintenance 79% effort

Because you need all of above mention stages for it.

longest time.

2 types of developers in big companies

Developers → Build requirements

in a time frame.

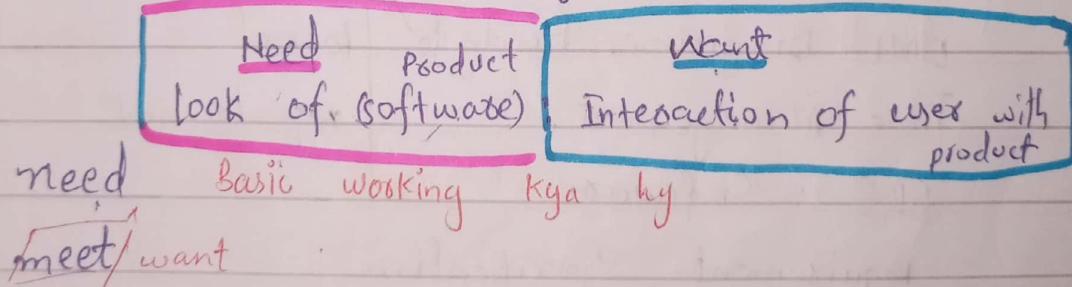
Deliver and move on

to next (time) task

support team → 24 hours working to maintain

Software can malfunction at any time.

Whereas at low level software houses high learning is, because developer, tester, and maintenance. high pays.



- Order kya kis kis phase sy guza
- How client/users interact with your software?

IEEE : International Organizations

ISO : International ^{standard} Organizations

They build standards

NetSole Level - 5

CMM CMM

Capa tells level of understanding

NetSole works on standards

like hiring, building, nikalna-etc

Def

Requirement:

"A condition or capability that must be met by a system or its components to satisfy a contract, specification"

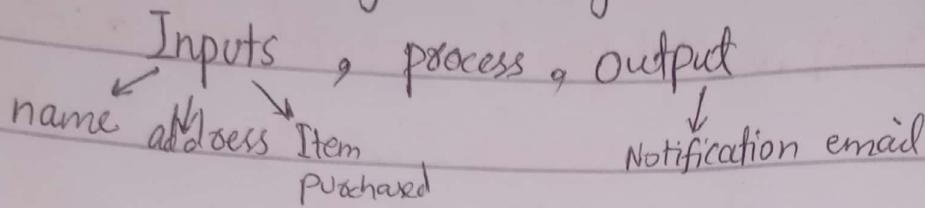
Specification : 8GB RAM) | or the
UML Diagrams, Budgets, | formal
Time Management, | requirement document
SRS : Software

Requirement Specification

Functional Req.

The main req of product is functional req.

User → login → Registered

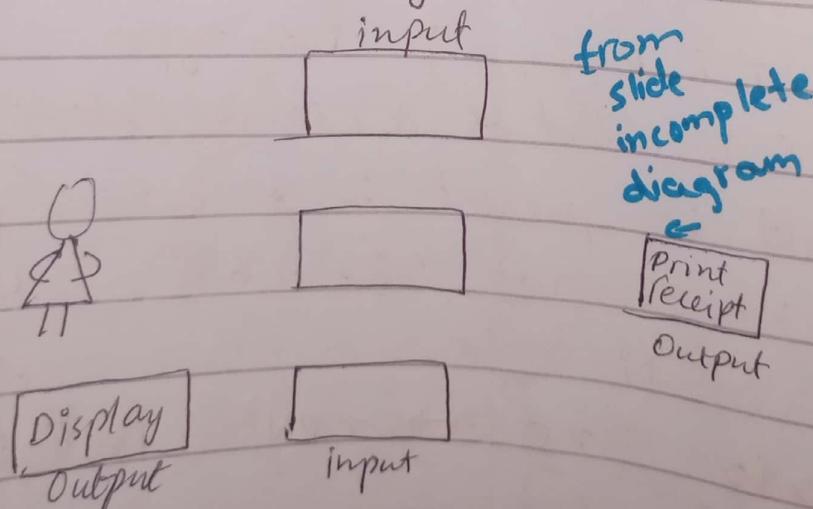


How to write:

As a user, I shall log into the system.
 Shall shows rigidity, stress and importance.
 Cells functional / non-fun
 always use shall → must

X should X → not to use instead of shall

Information flow diagram



Business Req.

percentages

revenue

profit

Don't mix it with business rules, constraints
two main constraints of software

↳ Time

↳ Budget

policies, guidelines, regulations
you can't misuse

CNIC, otherwise

legal actions

will taken

→ Legal regulations

→ Privacy

→ Brand uniformity (like taste of thing changes often
when their business grows)

→ .

Users / End users

User req. are functional ~~so we shall~~

func: sign in page, email notify

user can accomplish with product

wireframe: most time consuming task
and write down req. of each button.

- Use cases

- Story boards

- User stories

Non-functional req

Performance

Not okay \Rightarrow order \rightarrow email receive after 3 days

Not okay \Rightarrow 2 users $\xrightarrow{\text{allow}}$ to login at a time

How good, efficient and proficient is your software

As a user, I must log into the system
within after 3 sec.

exact method: Exact value of time, budget.

Range like: from 1 to 3 seconds.

if software take 5 sec, you have to
work \Rightarrow called test case.

main mistake: They don't consider ↑ test case
rapidly ⇒ how much time.

Response time: response of software in time +
The response time of software 'must' if (should
be (efficient) \rightarrow exact time / range
↓
bcz ← not okay ← shall
it is → non func.

Throughput (i.e. internet ki kisi Mbps use hoga,
functional | non-functional
: |
--- --- --- each point detailed
--- --- --- separately)

How much data is processed at a time?

How many users can use / place orders at a time?

Utilization

How many sources of system will be utilized? like 80% RAM, 20% Hard.

Scalability

e.g.: This system should be scalable tackle 10,000 user at one time. (future)

Capacity

Reverse of scalability: current.

Availability 24/7

Must be available 24/7

But you should explain (by ensuring
UPS. (generator).)

Reliability

if jeans purchased, shirt not be shipped.

Recoverability

System face problems/cash, it should be good enough to recover.
(Three types error) error must be removed by developer to recover identified and

Maintainability

75% works on it.

Serviceability

How much services are given at a time to how many users.

Security

unethical hacking, password, user name

Regulatory

?

Manageability

It must be easy to manage.

Our system must be manageable easy to understand graphics, text, button

like color schemes (For color blinds)

Environmental

Environmental friendly / Green environment
solar panel, electricity

Data integrity

Good security, data must be secure

Usability

Easy to understand, use

Interoperability

self-explanatory → show words ^{terms} on screen if

easy to use → guidelines, sub-sections

External tested
interface
Nadra, license management system.
whole profile CNIC

Physical requirements

product for undergraduates → need to be waterproof

Development constraints ?

Example:

Student user 1st access the homepage of the website
He sign in, can register sub, drop sub, view DMC,

functional:

- Our system shall offer students to sign in to perform different functions.
- Our system shall provide registration page to register subjects.
- There shall be option to drop subjects.
- Our System shall provide page to view DMC and attendance.
- Our System shall provide page to view attendance.

Non functional

① Security

- Our system must have username and password.
- Another student can't access others data.

② Performance

Availability: Our system must be available 24/7.

- Our system must be update information immediately. (1 to 3 sec) (Response time)
- Our system's processing must be fast. 1- to 2 sec.

③ Scalability

- Our system must tackle numbers of students (10,000) (In future).

④ Reliability

Information updated & calculated must be correct to reliable.

Registered or drop subjects must be correct.

⑤ Capacity \Rightarrow current users means present users 5,000 can be managed and tackled.

⑥ Users

Every student in the university.

⑦ User friendly
Use headings, Graphics, text, bulloons.

⑧ Throughput
How many data is passed through
5000 students must be registered at any
time

⑨ Capacity: current functionality (10,000 students
must be tackled).

⑩ Scalability: Future

Cost

⑪ Utilization: 90% of resources can be used

⑫ Volometric: How many data can handle

⑬ Availability: must be available, 24/7
by ensuring UPS.

⑭ Reliability: must be reliable, register sub
must be correct.

⑮ Recoverability: must be recover

⑯ Maintainability: must identify fault
so developer can remove it. (Error removal)

⑰ Serviceability: How many students can be cooked

⑱ Security: user name, secure from unethical
handling, physical security, stored in secure
place

1) Regulatory : Rules

2) Manageability : Use headings, coordinates can see registered sub.

3) Environmental: must not have hazard effect.

4) Data Integrity: Other app can't access our data.

5) Usability must be useable , text from screens, colorblindness. How system must be useable - How system is efficient.

✓ numbers should be defined

✓ No ambiguity

Vision: scope (value and scope)

the generic and short : (vague)

Our system will provide remote shopping
so students can shop remotely without
coming to store.

• purpose of product

should explain what is included as well
as what will not included.

Our system will provide functionality
to clients to buy products easily.

Our system shall have to functionality
where users can buy electronics from
their home using credit / debit car-

Vision, Scope, Func, non-func

what is common

Requirements

Scope creep ↴

Manager software fails ↴

Is this in scope. If not drop. If yes
talk to your client and discuss.

Stakeholders of a software

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

End users

Clients

(Prog-)Managers

Administrator

Types

| Primary | Secondary | Tertiary |
|--|--|---|
| end users who actually use the product | who occasionally use the product like youtube (Parent and child account) | who are affected by the product, make decisions about it. (company owners, managers etc.) |

User interface: UI user experience

Anything you can watch on software.

Screens, button, text, images

Challenges:

- ① User's inability to express what they need.

- ② Users are biased by previous bad decision.
- ③ Developers have trouble seeing through users point of view.

Solution: Always design for the intermediate level user.

Users limitations

- ① Perceptual / sensory limitation (color blindness)
 - ② Physical limitation (^{memory} left/right handed)
 - ③ Cognitive limitation (Memorize everything)
 - ④ Cultural limitation (Biased due to cultural h^{is}
word, buttons, are self explanatory that is the point)
like software for meat we can't use icons/multi media
against culture like dog, cat etc.)
- | | | |
|------------|----|-----------------|
| linux | VS | windows |
| ↓ commands | | ↓ user friendly |
| no virus | | virus |

How to gather information.

- ① Interview (UML, wireframes) in circle
- ② Focus groups (People in groups, all are writing reg.)
on page and forward to next, also write advantages / disadvantages)
- ③ Observing (Industry owners) & how everyone is working in industry, observe them & each and every functions and write them.

① Consulting previous products

① Questionnaire

10,000 users of industry & write questions
and spread in employees and

Gather req by sitting with user.

② Use cases

پہلی اور سال فریضیاً لبریز
2. اسے software کی
جیسے چیز کے use
کرنے والے کا
نکال دیں۔

③ Prototyping

(7.1.2.3.4)

Use Cases

- Used to gather requirements (want/need)
- used to write user requirements
- Apna dik sun ke perform karna
kyo us ke liye jo jo steps
follow kren, use cases hain.
Want to register? →
Open → login → Open button → register
related
- actors require for use case

Sign in → Buy product

Sign up → Create new account

to register → Students

to mark attendance → teachers

① Participating actors student / teacher

② Goals

What to achieve by performing UC.

③ Triggers

Pass to developer for implementation

when this function will perform is trigger.

when to use this use case in whole fu

④ Pre condition

device, internet, have account

all those condition that must be true

before (an action) execution of basic flow.

⑥ Post condition all conditions that must be true after ^{use case}_{basic} of flow.

Mistakes

⇒ Paragraph X

⇒ Always write in points ✓

Basic flow

1. Student will open a browser.

2. " " "

3. " " " Click on "sign in".

4. " " " Enter username.

5. " " " Enter password.

6. " " " Click sign in button.

Every function have unique use case.

Alternate flows

More than 1 method to perform a func.

If there is no alternate method.

Write "Not Available" (But must be include).

Exceptions
Problems while executing basic scenario.

Mistakes: Paragraph

Write into points ✓

① Student does not have account on LPA.s.

Solution:

→ Default coordinator will create ^{his} account

- ② Student does not have active internet.
- credit card has expired,
 - empty credit card.
 - forgot user name.
 - forgot password.

* Exceptions are important for test case.

* Developers write use cases.

* Testers write test cases.

(10) Qualities

non-functional req.

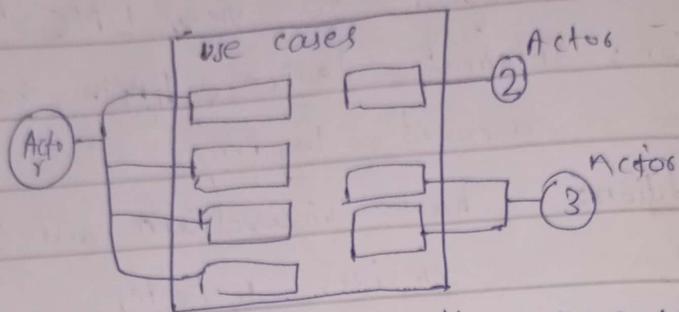
- System must not take more than 3 sec to login.

Example

- ① Name : Student View DMC
- ② Participating actor : Student
- ③ Goal : View DPMc
- ④ Triggers : Request to view DMC by clicking button
- ⑤ Precondition : Internet, device, ~~log in~~, have account, log in
- ⑥ Post-condition : have viewed GPA and CGP of previous semester
- ⑦ Basic flow : Have account → log in by Entering username and password → click on view DMC.
- ⑧ Alternate flows : Not available
- ⑨ Exceptions : → Students don't have account.
 - wrong username → Internet Problem,
 - wrong password → Not availability of ~~Internet~~ device
 - forgot username / password
- Solutions:
 - Create account
 - Enter correct password / username
 - Solve internet Problem
 - Have a device
- ⑩ Qualities:
 - System must be accessed within 3 sec.
 - System must be available 24/7 hours. by ensuring ~~correct~~ marks and gpa.
 - System must be updated regularly.
 - System must be secured. by login ~~username~~ & password.

Use case Diagram

Actors, use cases and their relation



first draw then write use case

Use cases are use

to identify and
write down ^{use} req.

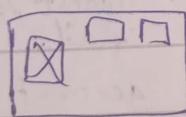
Wireframes

These are GUI
of user requirements.

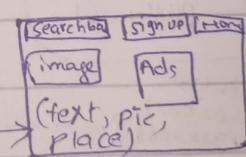
Graphical representation of User Req.

Mock-up: How the user system's
interface look like up.

Types



→ Detailed wireframe



→ Short/simple wireframe

only showing elements
place.

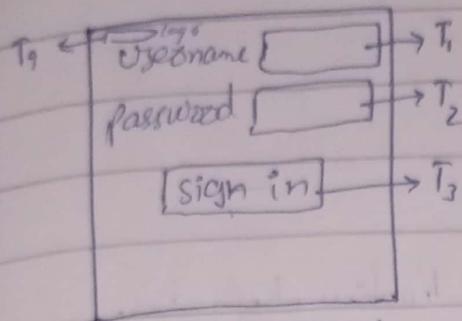


→ To demonstrate idea.

→ helping clients

→ to get feedback of user regarding
req so that he can be clear what
he wants and what he is saying

Want
Signed



use case

write into points

wireframe

Draw the screens.

T₁, T₂, T₃, T₄ keyword

Admin → have control of user

Every screen must have two roles user, admin

Table

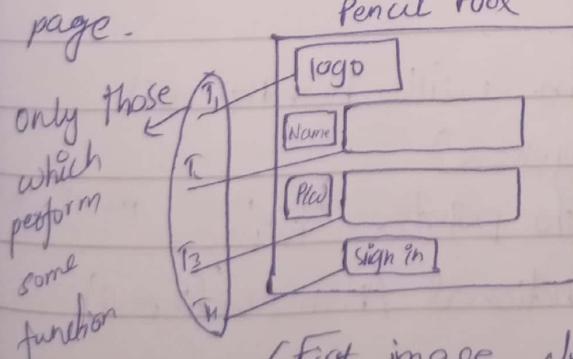
As a user
I shall enter username,
,, password
,, click sign in.

I shall click on
logo, move to home
page.

As a admin
I shall access/
modify / delete/
update the
username of
user.

I can change/
update logo

pencil tool



boundary

Keywords

T₁, T₂, T₃ must
be out of
boundary.

otherwise

X mask no.

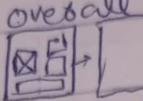
(First image draw
then table write)

SE Lee

Story Boards

just like wireframes

High level of user experience

Discussing overall functionality of System


Low level

Details of each detail with heading, colour, name etc. In depth

Persons

persons who are going to use software.
age, ethnicity, income level, job, personality

Activity diagram

- User click on sign in.
- Step perform by user is activity.

State Diagram

- After activity, step perform by system is state.

after every activity state is changed.

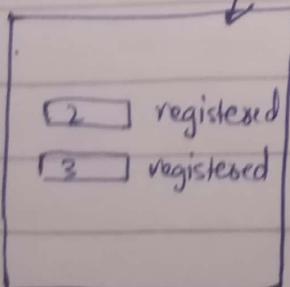
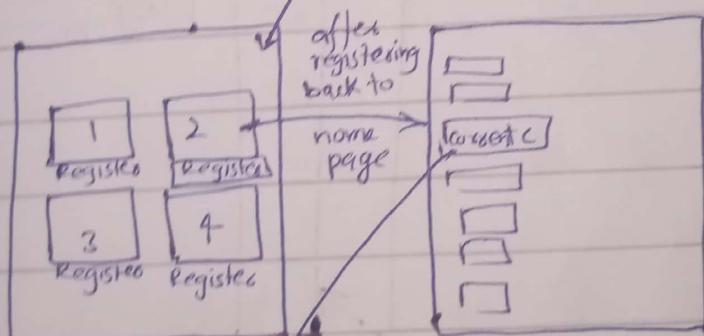
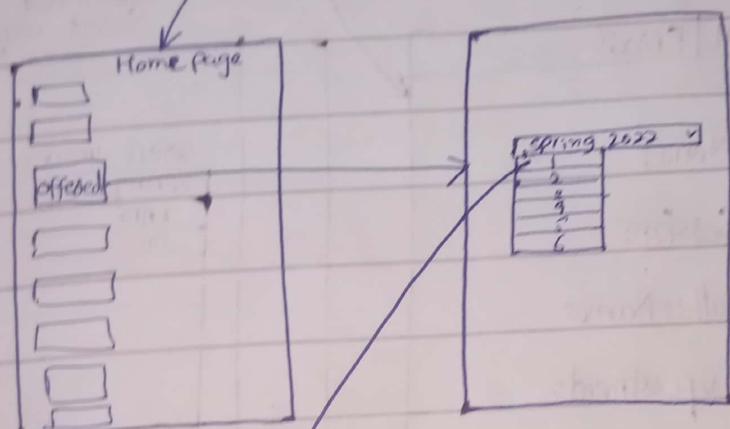
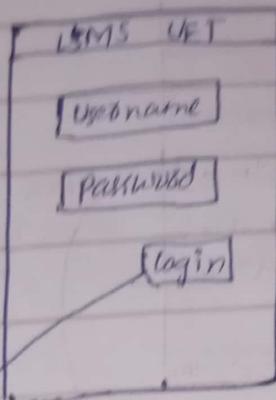
Example

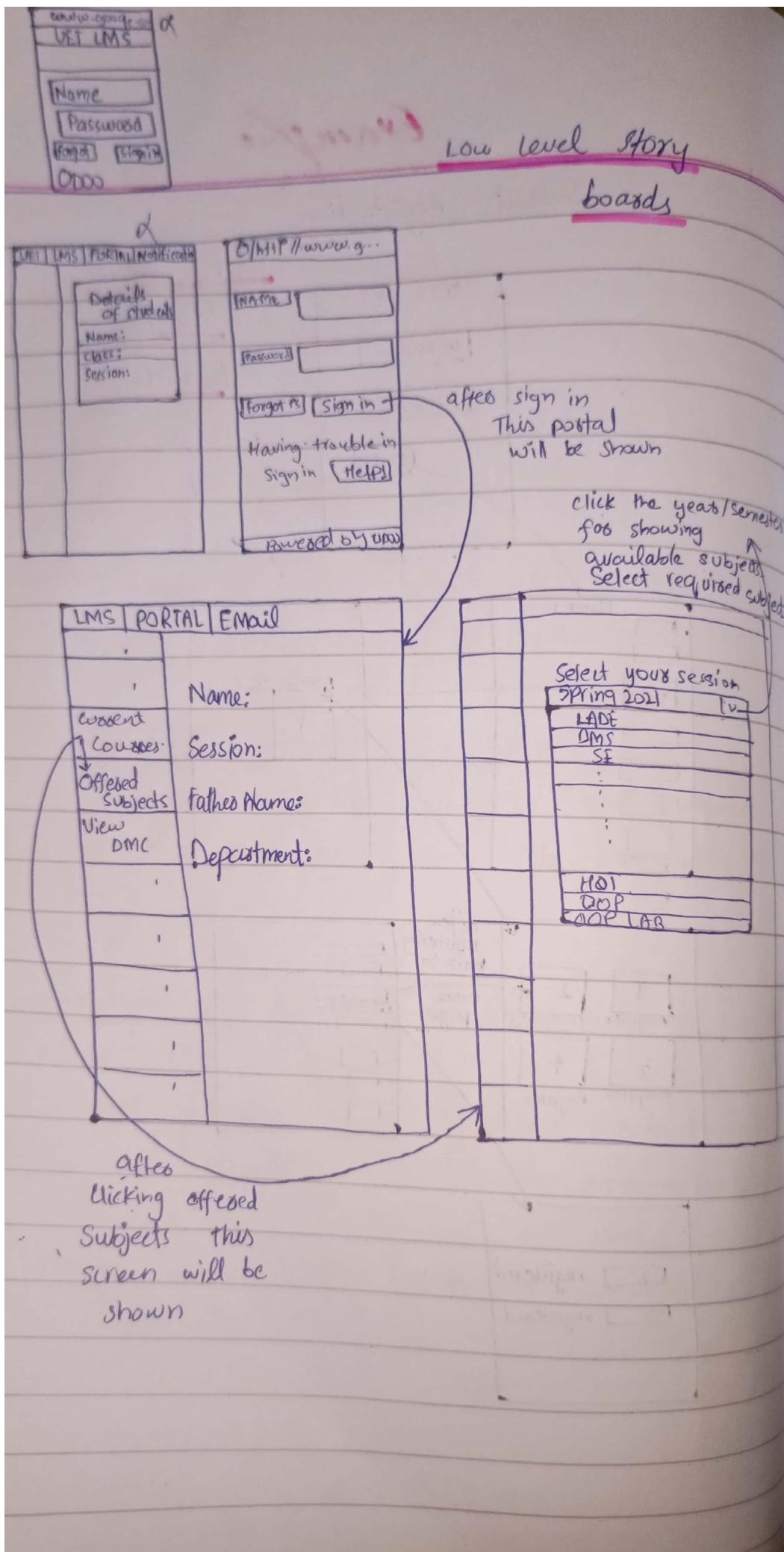
Subject registration

High level

Story

boards





| LMS | PORTAL | Email | | LMS | PORTAL | EMAIL |
|------------------|--|--|--|--|----------------------------------|--|
| Current courses | LADE teacher: section: | OOS: teacher: section: | after clicking registered go to | Current courses | Subject current Registered | |
| Offered subjects | Register | Register | | tab to view that have you registered or not. | 1 2 3 4 5 | Registered Registered Registered Registered Registered |
| View DMC | SE teacher: section: Registers | OOP teacher: Section: Registers | | | | |
| | QFT Teacher: Section: Registers | OOP LAD Teacher: Section: Registers | | | | |
| | | | | | | |
| | | | Powered by OOSO | | | Powered by OOSO |

- ① As a student so that I can access the browser ~~and~~ sign in my account.
- ② As a student, I want to go to offered subjects button so that I can register in my subjects.
- ③ As a student, I want to select required subject to register.
so that I can.
- ④ As a student, I want to check. I am registered or not, I shall click on ~~offered~~ subjects.

EPIC USER STORY: As a student, I want to register my subjects so that I can sit in class.

User Stories

In textual form, user requirements

who
what
why

As a user, I want to be able
to identify dietary restrictions, so that
I know I can eat the food that
I order.

- As a who, I want to what so that why
- Stakeholder: who has +ve/-ve effect or
may be affected by Software.
- Epic User Story
constraints budget and time
cone of uncertainty.
User stories → breakdown → pieces → then
calculate.
- User story uses & point of
view say banti my.

- How to develop software after software analyst have gathered requirements.

Stump model

Customer visits and checks the software working and suggest changing during development.

SRS

Software Requirement Specification

whole syllabus

12 Principles of Agile

(Last time
Question aya
tha)

① Early & continuous Delivery

جیسا کہ customer کو سب سے پہلے سے جائز نتائج دیا جائے

②

قابلِ software کو بھی جیسا کہ باقاعدہ کرنے والے جیسا کہ

③ Technical excellence

برنامروں کو excellent design کی وجہ پر بنانے

④

⑤ Self

کہ customer اور developer مابین change / job کو کم کر کر تو فیض، کم کر کر تو فیض کرنے والے

- ⑥ Delivers frequently
 - ⑦ Welcome changing Requirements
جذب تغييرات متطلبات البرمجيات
 - ⑧ Sustainable Development
 - ⑨ Build Projects around Motivated People
 - ⑩ Daily collaboration
 - ⑪ Reflect on team behavior

SE Lecture

SDLC

Planning

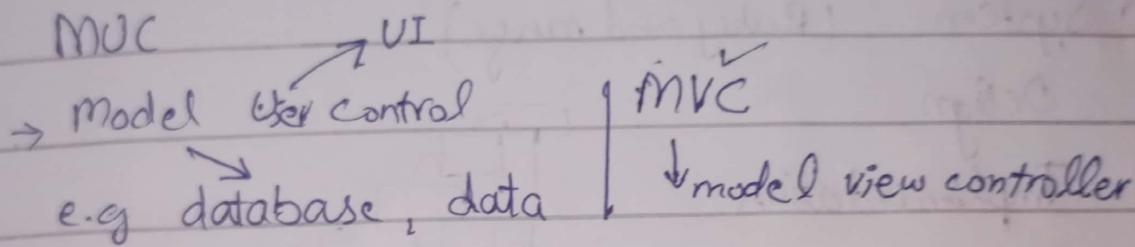
Requirements

Defining

What could be needed? why needed.

Designing

MVC



Talk about architecture

Building

Coding / implementation

Testing

Unit

System

Integration

Deployment

delivers it to users/ customers

Waterfall

(most ? o)

linear sequential

Basic and fast model

Every model have base of waterfall.

① Requirement Use story, use cases
Analysis

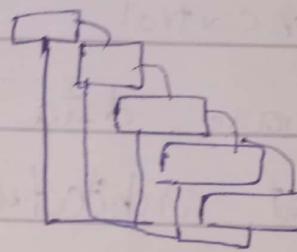
② System (Puty and range)
Design

③ Implementation

④ Testing

⑤ Deployment

⑥ Maintenance



logistics will be?

⇒ 2 times a person comes, clarifies the requirements, no need much change, and time come to take it.
need Diagram, Description of each step,
in paper Advantages and disadvantage.
Object-oriented class is small module

Iterative Model

smaller junks, prototypes to keep track what have been developed

Design and implementation

Testing

Implementation

Build

1 1

2 3

B2

1

2

3

B3

1

2

3

working model

working model

at the end

at each iteration

waterfall

Iterative

Spiral Model, Big Bang Model

small functionality

code it and deliver it. (No SRS signed)

Lab me mostly ye follow hota hy

Requirement analysis

specification, prioritize

try to identify, discover errors / risk analysis

before start up, even

Identification

Design

- module \rightarrow different parts of software.
- architecture \Rightarrow database.

Build

POC

FYP

Requirement analysis and risk analysis

to problems a riskti hoin software me
uh ko tackle kro.

Table \rightarrow SQL

adjacent files \rightarrow no query

Applications

medium to high level risk projects

V-Model

① Verification: Checking of all requirements have been developed.

② Validation: Checking if " " " are working fine.

Req
System
High: Architecture examin, administration all VET
fee ; combine
low: Module CS, main, national
 individual
test name: alphabets , special , char , numeric
check

@ . -
Integration unit ki combine , aur module ki
combine testing → system testing.

Aceptational cases

Testing according to user want,
requirements, using test cases

USER want

SDIC Model (PAD model)

Incomplete
just
Topic Names

Business Modeling / also called requirement gathering
How manufacturing

Data Modeling

Structure?

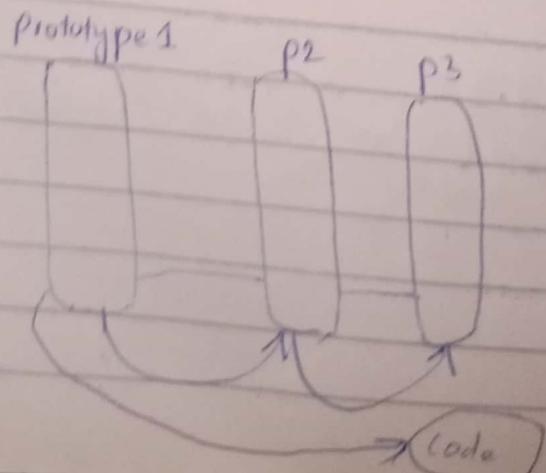
What we need to build that system
Reviewed and analyzed.

Process Modeling

Application generation

Loding

Testing and turnover



RAD vs traditional
changes N/A
rigid

iterative and incremental

Software Prototype Model

- limited functionality
- to communicate with clients

- 1 Gather requirements
- 2 Develop
- 3 Review prototype
- 4 Revise and enhance prototype

* Imp Question What are types of prototype
Open ended, Closed ended

Throwaway (close ended)
Rapid.

Evolutionary
breadboard prototyping
actual functional prototype with minimum functional

Incremental

Extreme Prototype
GUI banana hy first

Inf 

Skills

Implementation of Agile Model

Scrum: implementation of Agile

Mostly used in industry
team 3 to 9 members

Extreme programming (means 2 developers instead of 1)

Scrum Master: FYP Grp leader, actual manager of software, coach as manager.

Sprint: Divided work with time
pre sprint, post -.

Daily Scrum

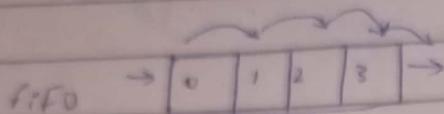
What mila, kr liya, next ye mila ga,
ya ye mila a rha hy

Project Planning

base on P. backlog.

Two meetings

Product backlog → sprint backlog → Sprint → small software
all software req.
those req. in 15 days next



light weight queue based
Product product backlog queue (user story)
Product Owner

Product Owner writes, prioritizes, add them to product backlog.

V. imp farmi ave ga → Artifacts: Sprint backlog
from
slides