Vivekanand Education Society's Institute of Technology

Department of Computer Engineering

Subject: SE

Class :- CMPN(D12)   Semester:- V   Div :- A

Roll No: 44	Name: Chirag Lundwani		
Exp No:	Title: Assignment 1		
DOP:	30-08-2021	DOS:	08-09-2021
GRADE:	LAB OUTCOMES:		SIGNATURE:

5] Explain cmm levels with examples:

Ans: A maturity level is a well-defined evolutionary plateau towards achieving a mature software process.
Each maturity level provides a layer in the foundation for continuous process improvement. In cmm models with a staged representation, there are 5 maturity levels designated by numbers 1 to 5.

I] Level 1: Initial
- → At level 1, the process is usually chaotic & adhoc.
- → A capability is characterized on the basis of the industrial & not on organization.
- → Progress isn't measured.
- → Products developeds are often scheduled & over budget.
- → wide variations in the schedule, cost, functionality. & quality targets

II] Level II: Managed.
- → Requirement management.
- → Estimate project parameters like cost, schedule, & functionality measure actual progress.
- → Develop plans & process
- → Software project Standards are defined
- → Identify & control products, problem reports changes etc

III) Level III: Defined.
- → clarify customer requirement.
- → Solve design requirement, devlop an implementation process.

→ Analyze decisions systematically
→ Rectify & control potential problems

IV) Level 4: Quantitatively managed.
  → manages the project's processes & subprocesses. statically
  → understand process performance, quantitatively manages organization's project.

V) Level 5: Optimizing:
  → Detect & remove cause of defects early.
  → Identify & deploy new tools. & process improvements.

Q.3) Explain pros & cons of using a waterfall model.

Ans: waterfall model is described as a linear sequential lifecycle model. Here, the result of each phase cascade down to the next level of development.

Pros:
→ Timescales are Kept
→ Everyone gets up to speed quickly
→ No financial surprises
→ Testing is made easy
→ Outcome is crystal clear

Cons:
→ Needs can be difficult to define
→ Potential lack of flexibility
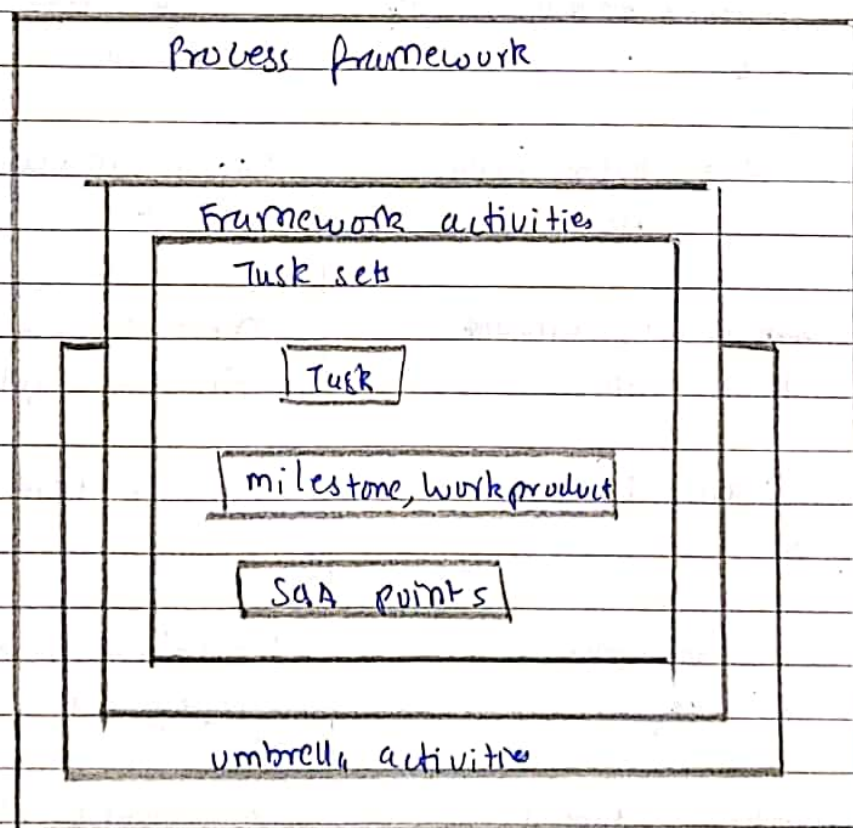→ Longer delivery time.

Q.31   Give the comparison between spiral model & RAD model

RAD model	Spiral model
RAD model is a software devolopment model whereby the components or functions are developed in parallel as if they were mini projects	Spiral models is a software development model & is made with features of incrementally, waterfall or evolutionary prototyping
RAD model requirements & early stage planning is not necessary	spiral model requirements & early stage planning is required
RAD is used between large & small projects	Spiral model is used for large projet.
There is low amount of risk in RAD	There is low amount of RisR in spiral
Small team size is required	large team size required.
Flexibility to change in RAD is easy	Flexibility to change in spiral is difficult
overlapping of phases is possible	overlapping of phases is not possible.

Q.4) Illustrate framework & umbrella co activity

Ans:- Framework:-

It is a standard way to build & deploy applications. Software process framework is a foundation of complete software engineering process. Software process framework includes all set of umbrella activities. It also includes number of framework activities that are applicable to all software projects.

A generic framework encompasses five activities.

```
┌─────────────────────────────────────────────┐
│  Process framework                          │
│   ┌────────────────────────────────────┐    │
│   │  Framework activities              │    │
│   │   ┌─────────────────────────────┐  │    │
│   │   │  Task set                   │  │    │
│ ┌─┤   │   ┌──────┐                  │  ├─┐  │
│ │ │   │   │ Task │                  │  │ │  │
│ │ │   │   └──────┘                  │  │ │  │
│ │ │   │  ┌──────────────────────┐   │  │ │  │
│ │ │   │  │ milestone, workproduct│  │  │ │  │
│ │ │   │  └──────────────────────┘   │  │ │  │
│ │ │   │   ┌───────────┐            │  │ │  │
│ │ │   │   │ SQA points│            │  │ │  │
│ │ │   │   └───────────┘            │  │ │  │
│ │ │   └─────────────────────────────┘  │ │  │
│ │ │  umbrella activities               │ │  │
│ └─┴────────────────────────────────────┴─┘  │
└─────────────────────────────────────────────┘
```

i) **Communication:**
In this activity, heavy communication with customer & other stakeholders, requirement gathering is done.

ii) **Planning:**
In this activity, we discuss the technical related tasks, work schedule, risks etc.

iii) **Modelling:**
It is about building representations of things in the real world. In this, a product's model is created in order to better understand requirements.

iv) **Construction:**
In SE, construction is the application of set of procedures that are needed to assemble product.

v) **Deployment:**
In this, a complete or non-complete product or software is presented to the customer to evaluate & give feedback, on basis of which we modify project.

umbrella activities in a software process are:-
i) Project tracking & control
ii) Formal technical reviews
iii) Software quality assurance
iv) Software configuration maintainence
v) Document preparation & production.
vi) Reusability management

Q.5) What do you understand by agile methology?. Explain details XP, Scri scrum, & kanban mode.

Sol^n: Agile model-: methology:

It is a management paradigm used in software deve development environments. Agile has been used seen massive adoption over the past decade. 71% of organizations surveyed reported that they were used using agile methology. Agile emphasizes work in short iterations & frequent releases of working software, with a fast feedback loop back to developers regarding actual values.

Scrum:

It is a highly iterative agile framework that operates in sprints of 2-4 weeks. It defines features & objectives prior to each sprint, & is designed to reduce risk while providing value to customers as quickly as possible. In each sprint, the team commits to completing several user stories - brief descriptions of what a user needs to be able to achieve with the software.

Product backlog   Sprint log   Potentially shippab. product Increment
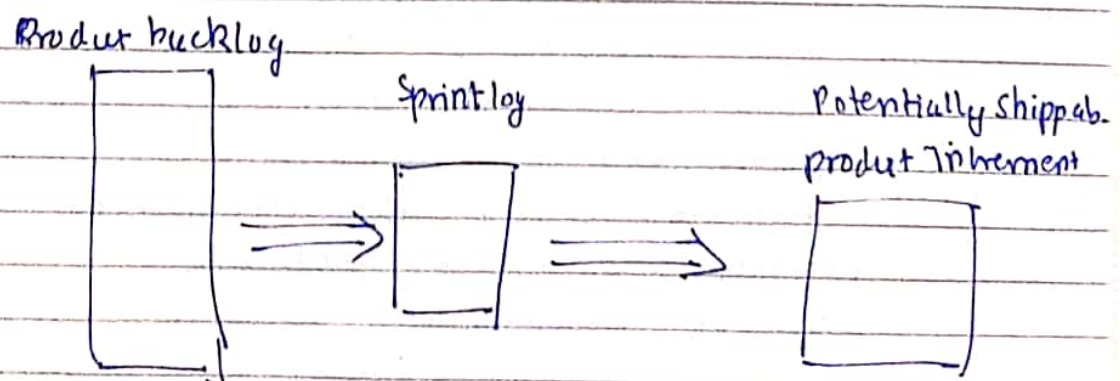
The primary stages in the scrum process are:

I) Creating a product backlog:
A prioritized list of development tasks, defined as user stories. The work required for each user story is estimated using hours or story points

II) Sprint planning:
creating a sprint ~~back~~ backlog - a subset of the product backlog planned for a specific sprint & estimated to fit into the fixed time scope.

III) Sprint work: Developing working software within the sprint. The team ~~team~~ carries out a daily stand up meeting to share progress & resolve problems

IV) ~~Rest Retr~~ Testing & product demo:
Towards the end of sprint, the focus shifts to the stabilizing f.& finalizing features, & conducting acceptance testing

V) Retro spective:
At the end of sprint, sharing lessons learned from previous sprint & using them

Kanban:
It is an agile -based methology, which originated from lean ~~mmol~~ manufacturing, pioneered by Toyota. It is a large todo list, which helps manage work according to priority. A central principle of Kanban is that the tasks & their status are visualized as cards on a board, visible to all. In a Kanban, whensomeone is ready for more work, they pull a task from the board by moving it to doing or a status like Testing

The extreme programming:

It is a highly disciplined management method, which focuses on continuously improving quality & speed of software delivery. The dev team works closely with customers, continuously planning, testing & providing feed back to developers, to quickly deliver variable valuable software.

The lifecycle stages:-

i) Planning
ii) Designing
iii) Coding
iv) Testing

Q·6) Explain requirement engineering process in detail.

Ans: It is the process of defining, documenting & maintaining the requirements. It is a process of gathering & defining service provided by the system. Requirement process consists of defining of the following main activities:

I) Requirement Elicitation!

It is related to the various ways used to gain knowledge about the project domain & requirements. The various sources of domain knowledge include customers, business manuals, the existing software of the same fin type, standards & other stakeholders of the project.

II) **Requirement specification:**
This activity is used to produce formal software requirement models. All the requirements including the functional as well as the non-functional requirement & the constraints are specified by these models in totality. During a specification, more knowledge about the problem may be required which can again trigger elicitation process.
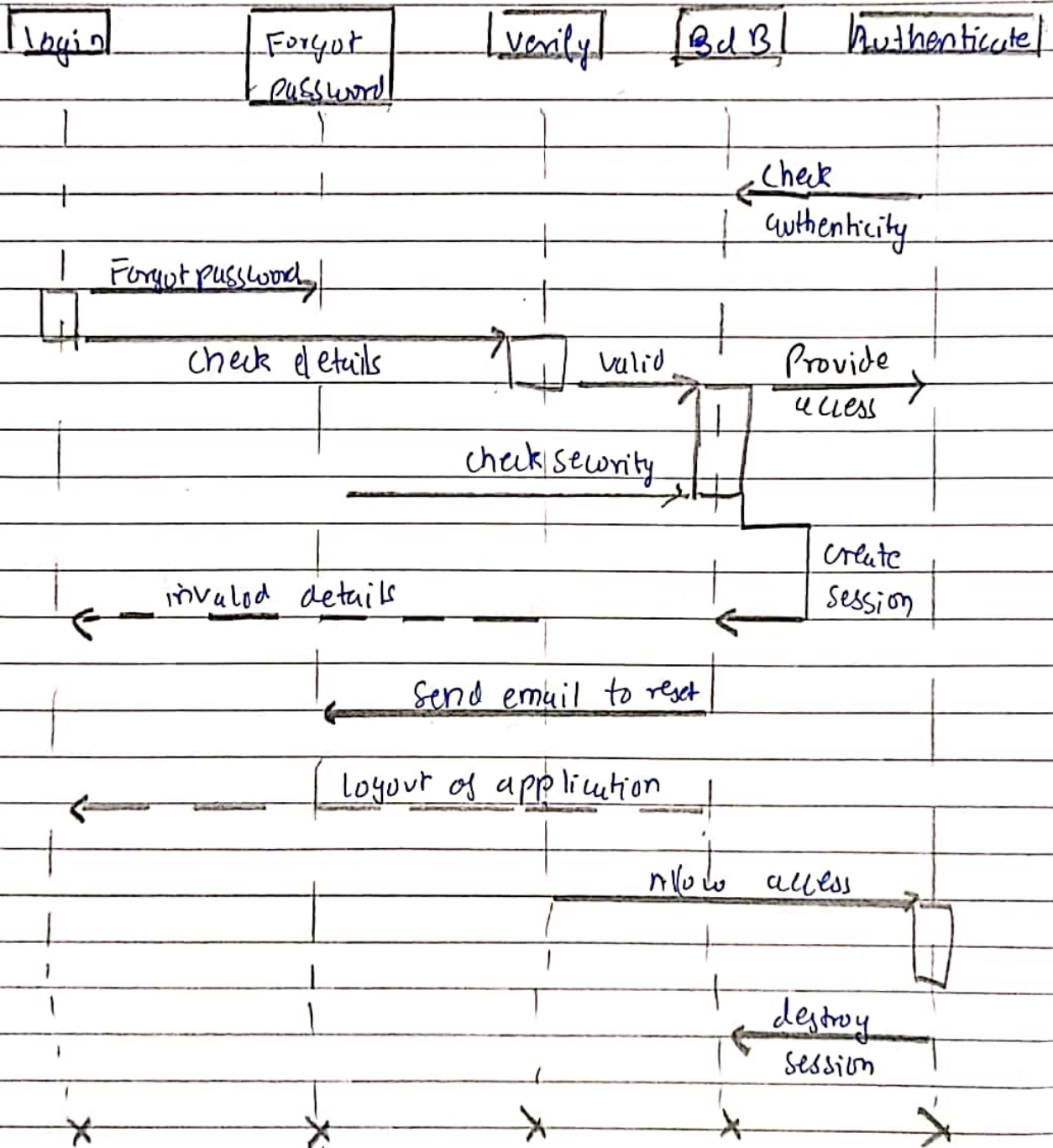
III) **Requirement verification & validation:**
verification refers to the set of tasks that ensure that the software that has been built is correctly implementing a specific function
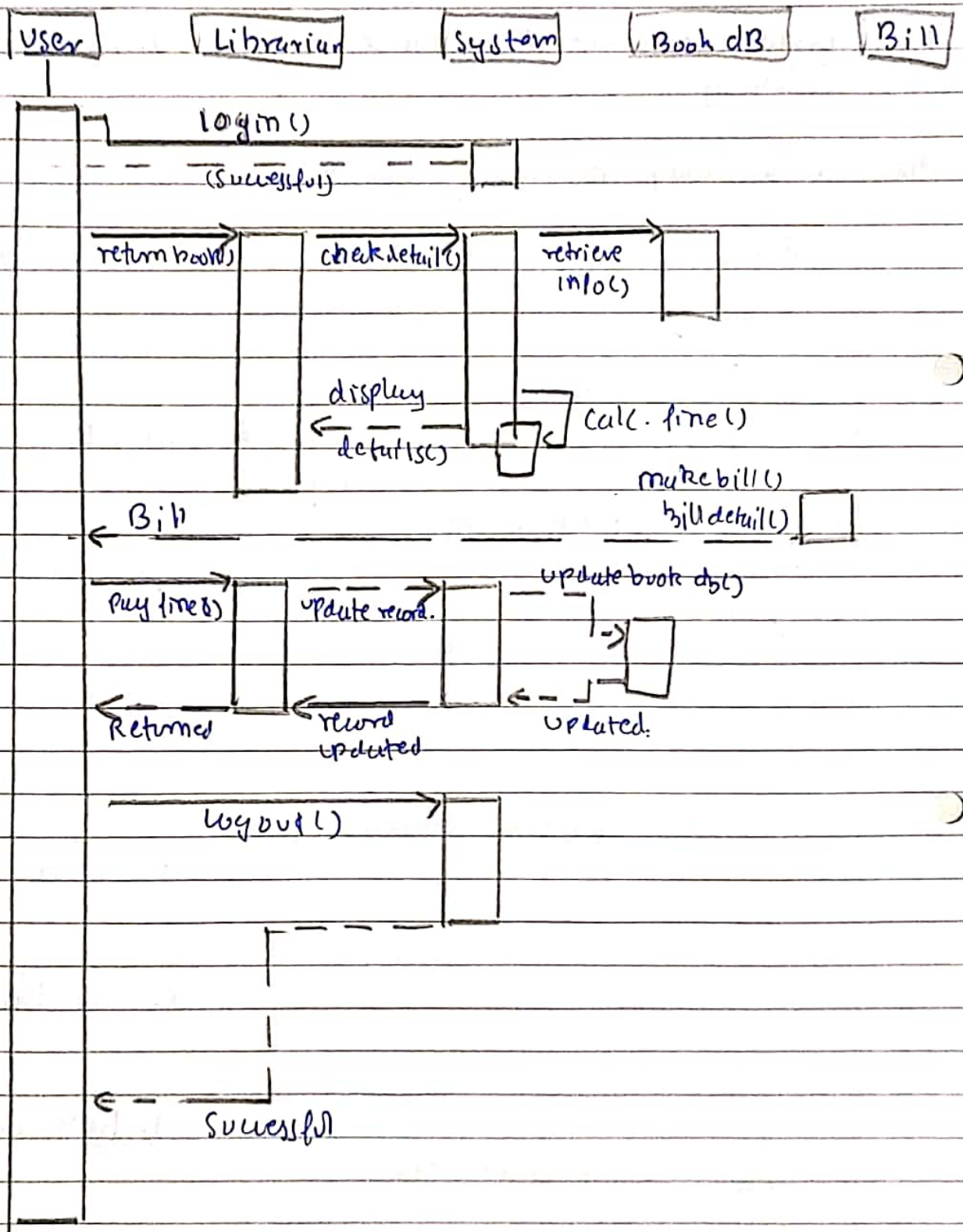validation refers to a different set of tasks that ensure that the software that has been built is traceable to customer requirements. If they aren't validated, errors would propagate to successive stages

IV) **Requirements management:-**
It is the process of analysing & documenting, tracking, prioritizing & agreeing on the requirement & controlling the communication to relevant stakeholders. This takes care of the changing nature of requirements. It should be ensured that SRs is a modifyable as possible so as to incorporate changes as in requirement specified by the end users at later stages too. Being able to modify the software as per requirements in a systematic & controlled manner is an extremely important part of the requirements engineering process
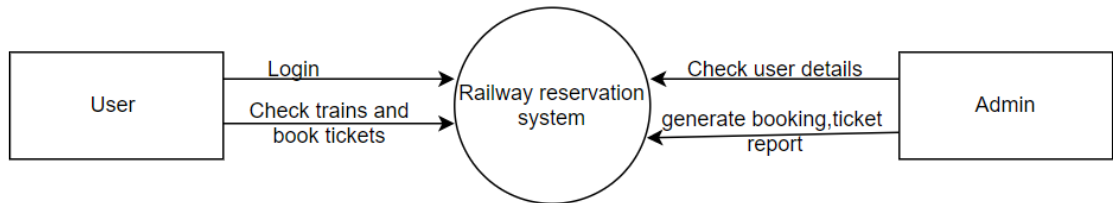
(ii)   Hostel management:

| login | Forgot password | verify | Bd B | Authenticate |

Check authenticity

Forgot password

check details → valid → Provide access

check security

Create session

← invalod details

← Send email to reset

← logout of application

n/o w access →

← destroy session

Sequence Diagram

User	Librarian	System	Book dB	Bill

login()

(Successful)

return book()   check detail()   retrieve info()

display details()   Calc. fine()

make bill()
bill detail()

Bill

update book db()

Pay fines()   update record.   updated.
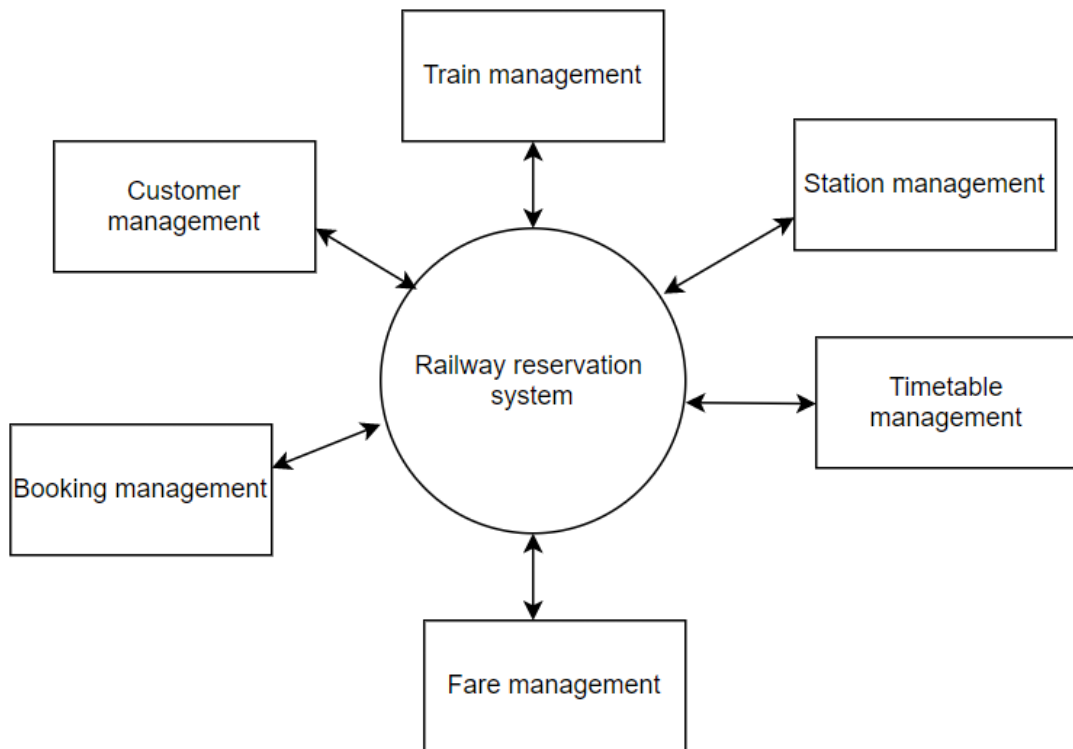
Returned   record updated

logout()
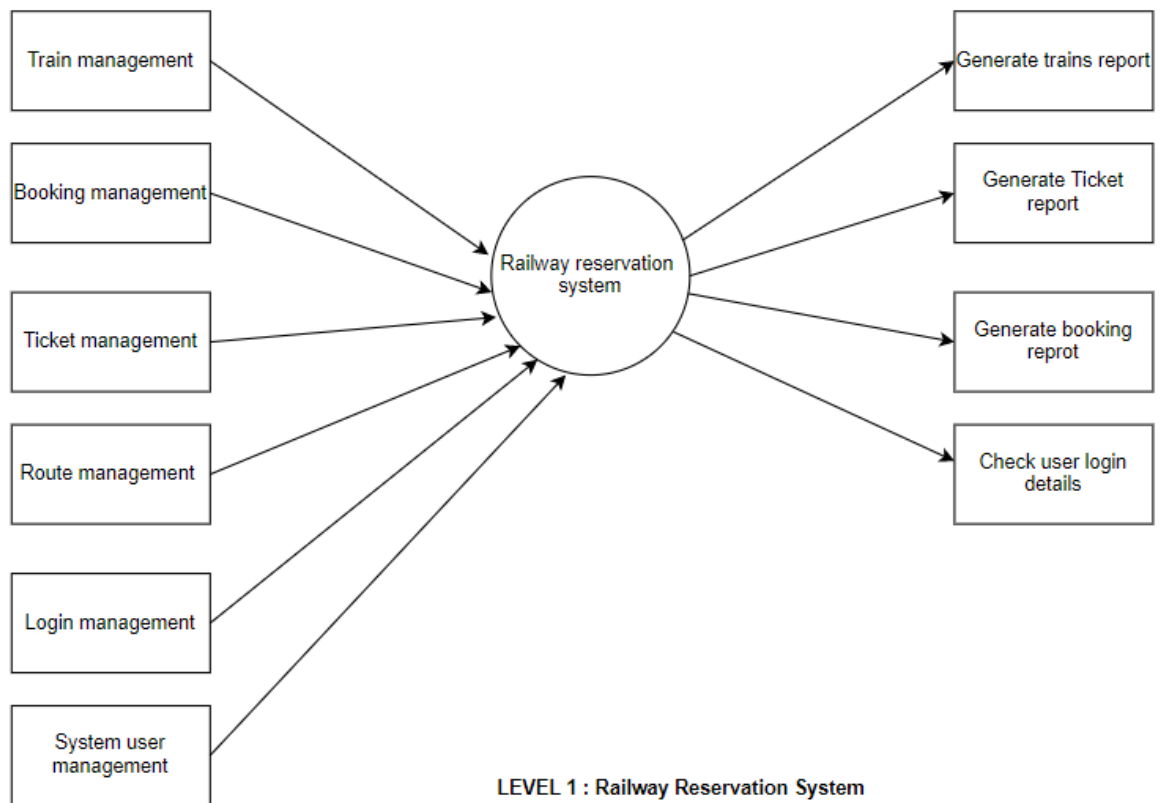
Successful

DFD diagrams:

1) Railway reservation system

Context Diagram: Railway reservation system

LEVEL 0 : Railway Reservation System

Train management

Booking management

Ticket management

Route management

Login management

System user
management

Railway reservation
system

Generate trains report

Generate Ticket
report

Generate booking
reprot

Check user login
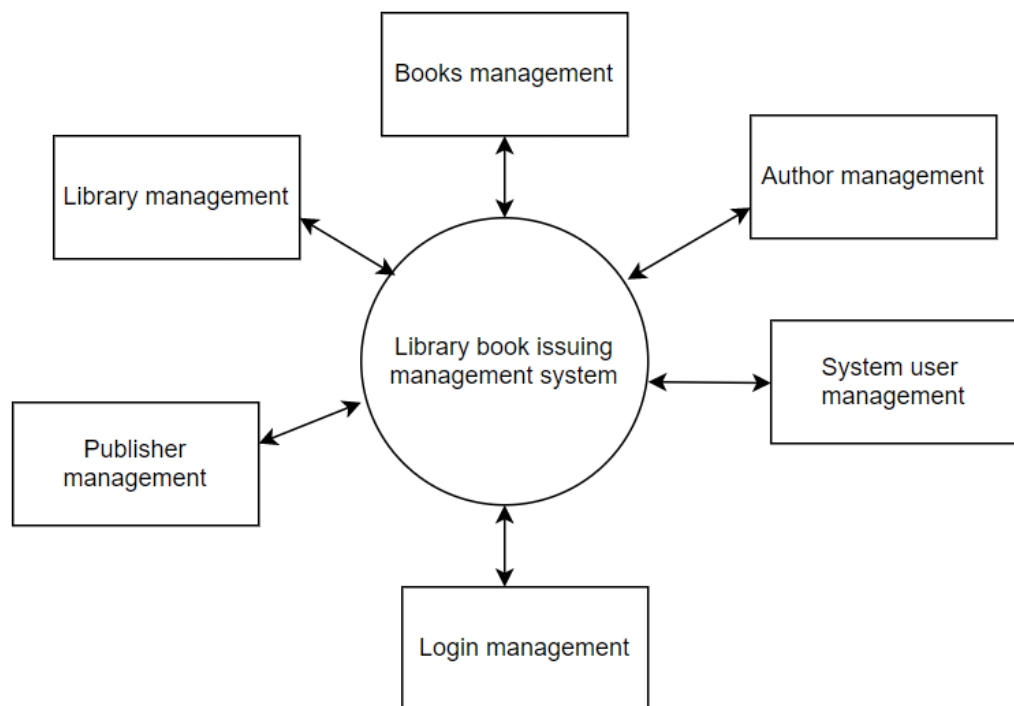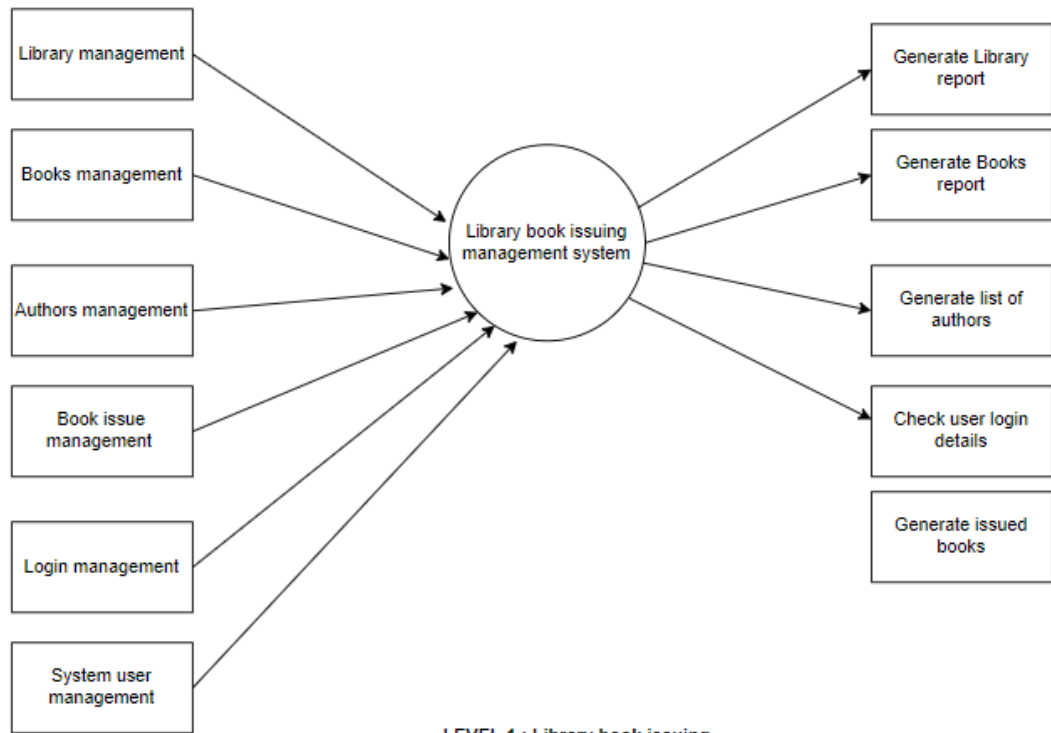details

LEVEL 1 : Railway Reservation System

2) Library book issuing management system
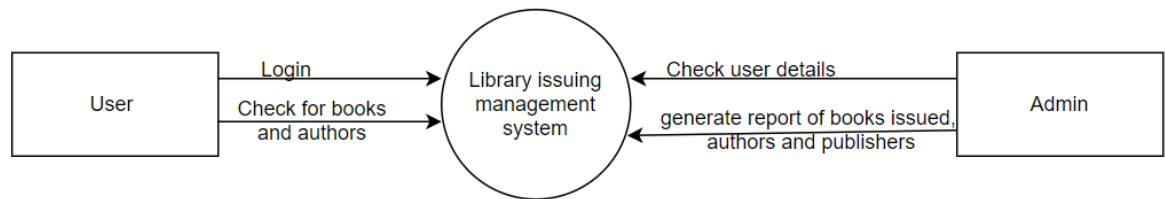
Context Diagram: Library issuing management system

LEVEL 0 : Library book issuing management system

Library management

Books management

Authors management

Book issue
management

Login management

System user
management

Library book issuing
management system

Generate Library
report

Generate Books
report

Generate list of
authors

Check user login
details

Generate issued
books

LEVEL 1 : Library book issuing
management system