

READ THIS FIRST

Remove this page before you prepare your own synopsis.

What this file is

A study reference sample of a BCSP-165 project synopsis for the IGNOU BCA_NEW / BCA_NEWOL programme. It is built heading by heading on Annexure-2 ("Suggested Template for Project Proposal/Synopsis") and Section III of Project Guidelines: BCSP-165 (School of Computer and Information Sciences, IGNOU, July 2026). It is published by ignouproject.com, an independent service not affiliated with or endorsed by IGNOU. It is not an IGNOU document.

What it is not

It is not a synopsis you may submit. The guidelines require genuine, original work, and copied synopses and reports receive very low scores and may be sent to the Unfair Means Committee of IGNOU (Section V). Choose your own problem, gather your own requirements, and draw your own diagrams. Every fact in [square brackets] can only come from you: centre name, city, learner and seat numbers, dates, guide details and your enrolment number.

Where Section III's eight items sit in this sample

Section III item	Where you will find it
1 Title of the Project	Title page
2 Introduction and Objectives	1.1 Background, 1.2 Objectives
3 Project Category	1.5 (Annexure-2 has no heading for it)
4 Analysis: DFDs at least up to second level, ER diagram, database design	2.3 (Figures 2.1 to 2.9), 3.2
5 Complete structure: modules, data structures, process logic, testing process, reports	3.1 (Tables 3.1, 3.2), 3.2 (Table 3.3), 3.3, 3.6
6 Tools, platform, hardware and software requirements	1.4 (Table 1.1)
7 Industry or client: Yes or No, with name and address	1.5
8 Future scope and further enhancement	3.7 (Annexure-2 has no heading for it)

The mapping rule this sample demonstrates

Annexure-2 ends with three notes: map ER entities to DFD data stores and to database tables; map project modules to first-level DFD processes; extend every first-level process to the second level. Table 2.3 shows that mapping for this sample: six modules, six Level-1 processes, eighteen Level-2 processes, six data stores, six entities, six tables.

Length and layout

The proposal checklist asks for 12 to 15 pages (p. 60). Chapters 1 to 3 and the References here run to 15 pages; front matter carries roman numerals. Layout follows Section 2: A4, Times New Roman 12 pt (Liberation Serif, a metric equivalent, here), left margin 2 inches, other margins 1 inch, page numbers centred at the bottom, numbered references in order of use. Section 2.5 sets single spacing for "the project proposal and report" and no other rule covers the synopsis, so this sample is single spaced.

What goes in the envelope (guidelines p. 60)

The proforma (p. 59) signed and dated by you and your guide; the guide's bio-data, signed and dated; this synopsis; a self-addressed stamped envelope. Send it to the Regional Director of your Regional Centre in the 1 April to 30 June or 1 October to 31 December slot, and keep a photocopy of the complete package.

Want your own draft checked against these rules?

ignouproject.com/ignou-project-synopsis-correction/ checks a learner-written synopsis or report against the BCSP-165 guidelines. You remain the author; IGNOU's evaluator decides approval.

INDIRA GANDHI NATIONAL OPEN UNIVERSITY**PROJECT PROPOSAL (SYNOPSIS)****LAB SLOT BOOKING AND PRACTICAL
ATTENDANCE TRACKER FOR A COMPUTER
TRAINING CENTRE**

by

[Student's Full Name]

Enrolment No.: []

Under Guidance of

[Guide's Full Name]

Submitted to the School of Computer and Information Sciences
in partial fulfilment of the requirements
for the degree of

Bachelor of Computer Applications

PROJECT COURSE (BCSP-165)

Indira Gandhi National Open University
Maidan Garhi
New Delhi - 110068

Regional Centre: [] Month and Year: []

Layout of this page follows Appendix 1 (Sample Title Page), guidelines p. 72. No
university logo is reproduced in this sample.

ORIGINAL COPY OF THE PROFORMA OF BCA_NEW/BCA_NEWOL PROJECT PROPOSAL

[Place here the original proforma printed on p. 59 of the guidelines, filled in, signed and dated by you and by your guide. A photocopy is not the original.]

Check before you place it here

- items 1 to 6 are all filled, including your e-mail and telephone number;
- item 2, the title, is exactly the title on the title page; the report title must later match it (Section VI, point 7);
- item 4: the guide's degree is in Computer Science or IT, and the guide is not guiding more than 8 BCA_NEW/BCA_NEWOL students;
- item 6, software used, matches section 1.4 of this synopsis;
- both signatures carry a date; the Office Use box is left blank.

The proforma also prints three notes: refer to the list of tools in Section VII, demonstration or execution of the project at the viva is a must, and synopsis comments must be implemented in the report.

CERTIFICATE OF AUTHENTICATION OF WORK

[Place the signed certificate here.]

Annexure-2 lists a "Certificate of Authentication of Work" in the synopsis front matter, but the BCSP-165 guidelines print no separate format for it. The nearest printed format is the Certificate of Originality on p. 70, written for the project report. Ask your guide or Regional Centre which wording to use; do not copy a certificate from another learner's file.

SAMPLE FOR STUDY REFERENCE,
NOT FOR SUBMISSION
ignouproject.com

ABSTRACT

Computer training centres with a fixed number of lab systems often allot practical slots through paper registers or messaging groups, which can lead to double bookings, idle seats and attendance counted by hand at the end of each month. This project proposes LabSlot, a web application through which learners book seats in lab slots published for their batch, instructors mark attendance only against confirmed bookings, and administrators see seat utilisation and course-wise attendance percentages. The system is planned as six modules that correspond one to one with the six first-level processes of its data flow diagram and with six database tables. It will be built with Java 17, Spring Boot and MySQL 8, with hashed passwords and role-based access. The proposal covers requirements, conceptual models, database and procedural design, interface design and planned test cases for [Centre Name], [City].

Annexure-1 and Section 1.4 of the guidelines: one or two short paragraphs, 100 to 150 words. This abstract has about 130 words.

ACKNOWLEDGEMENT

[Name only the people who actually helped. For example: the student thanks [Guide's name, designation and organisation] for guidance in framing this proposal, and the staff of [Centre Name], [City] for explaining the present method of allotting lab slots.]

TABLE OF CONTENTS

Original Copy of the Proforma	ii
Certificate of Authentication of Work	iii
Abstract	iv
Acknowledgement	iv
Table of Contents	v
1. Introduction	1
1.1 Background	1
1.2 Objectives	1
1.3 Purpose, Scope, and Applicability	1
1.3.1 Purpose	1
1.3.2 Scope	2
1.3.3 Applicability	2
1.4 Software and Hardware Technologies (to be used)	2
1.5 Project Category and Client Details	2
2. Requirements and Analysis	3
2.1 Problem Definition	3
2.2 Requirements Specification	3
2.3 Conceptual Models	4
3. Basic System Design	9
3.1 Basic Modules	9
3.2 Database Design	9
3.2.1 Schema Design	9
3.2.2 Data Integrity and Constraints	10
3.3 Procedural Design	11
3.3.1 Flow Charts	11
3.3.2 Algorithms	12
3.4 User Interface Design	13
3.5 Security Issues	13
3.6 Test Cases Design	13
3.7 Future Scope and Further Enhancement	14
References	15

1. INTRODUCTION

1.1 Background

Many computer training centres and college labs have fewer computers than learners, so practical work runs in timed lab slots. At [Centre Name], [City], about [number] learners in [number] batches share [number] systems in [number] lab rooms; these figures will be confirmed with the centre during requirement gathering. Slots are at present announced on [a notice board / a messaging group] and seats are claimed in [a paper register].

Set-ups of this kind commonly face three problems, which will be verified at [Centre Name] before design is frozen: two learners claiming the same seat; seats left empty when a learner who claimed one did not come; and attendance totals compiled by hand. The proposed system, called LabSlot in this synopsis, moves slot publication, seat booking and attendance marking into one web application backed by a relational database, so that every seat, booking and attendance mark exists as exactly one record. The work involves software development through all stages of the software development life cycle, as the BCSP-165 guidelines require [1].

1.2 Objectives

The objectives of the project are:

- (A) to let learners view the open lab slots of their own batch and book or cancel a seat;
- (B) to prevent double booking and overbooking through database constraints and transaction-level checks;
- (C) to let instructors publish slots without room clashes and mark attendance against confirmed bookings;
- (D) to compute course-wise practical attendance percentages and flag learners below a threshold set by the centre;
- (E) to give the administrator slot-utilisation and attendance reports that can be exported;
- (F) to protect accounts with hashed passwords, HTTPS and role-based access.

1.3 Purpose, Scope, and Applicability

1.3.1 Purpose

LabSlot is intended to give each learner a fair and visible way to claim a lab seat, give instructors an attendance sheet generated from confirmed bookings, and give the centre accurate utilisation and attendance figures without manual counting.

1.3.2 Scope

In scope: three roles, six modules (Table 3.1), one centre with one or more lab rooms, browser access, CSV export.

Out of scope: fee collection, online examinations, biometric or RFID devices, SMS gateways, multi-centre use (see 3.7).

1.3.3 Applicability

Computer training institutes, college and school computer labs, and any organisation that shares limited equipment in timed slots.

1.4 Software and Hardware Technologies (to be used)

Table 1.1 lists the planned tools; Section VII of the guidelines asks learners to prefer the latest ones. Spring Boot 3 requires Java 17 or later [2].

Table 1.1: Planned software and hardware

Layer	Technology (planned)	Reason for choice
Language	Java 17 (LTS)	Object-oriented; studied in MCS-206; required by Spring Boot 3
Framework	Spring Boot 3.x: Spring MVC, Spring Data JPA, Spring Security	Layered web application with built-in authentication support
User interface	Thymeleaf templates, Bootstrap 5	Server-rendered pages with a responsive layout for phones
Database	MySQL 8.0 (InnoDB engine)	Foreign keys, CHECK constraints, row-level locking
Build, IDE, testing	Maven, Git; IntelliJ IDEA Community or Eclipse; JUnit 5, Mockito	Builds, change history, unit and integration tests
Operating system	Windows 10/11 or Ubuntu 22.04 LTS (development); any OS with a current browser (users)	Common, freely available
Hardware (development)	64-bit processor, 8 GB RAM, 20 GB free disk [state your own machine]	Runs the IDE, MySQL and the application together
Hardware (deployment)	[server or cloud virtual machine to be decided with the centre]	Depends on the number of users

1.5 Project Category and Client Details

Section III asks for these two items; Annexure-2 has no heading for them.

Project category: RDBMS (a web application on a relational database).

Done for an industry or client? [Yes / No]; if yes, [name and address]. If it later refuses to let you attach code, Section V needs its Project Manager's letter in the report.

ignouproject.com

Sample synopsis for study reference only. Write your own; do not submit this file. |
WhatsApp +91 93546 37830

2. REQUIREMENTS AND ANALYSIS

2.1 Problem Definition

The centre needs a single system in which (a) a lab slot can be published only once per room and time; (b) a learner can claim only a free seat in a slot of his or her own batch; (c) attendance can be recorded only for a learner who held a confirmed booking; and (d) attendance percentages and seat use can be reported at any time without manual counting. The present method, [describe the method observed at the centre], cannot enforce these rules because the register and the messages are not linked records.

2.2 Requirements Specification

The requirements follow the structure of IEEE Std 830 for software requirements specifications [3]. Table 2.1 lists the functional requirements with the module that serves each one; Table 2.2 lists the non-functional requirements.

Table 2.1: Functional requirements

ID	Functional requirement	Module
FR-01	The administrator creates, edits and deactivates user accounts with one role each.	M1
FR-02	Every user logs in with e-mail and password and can log out.	M1
FR-03	The administrator adds and edits courses (code, title, lab hours).	M2
FR-04	The administrator creates batches for a course and assigns one instructor to each.	M3
FR-05	The administrator enrolls each learner in one batch.	M3
FR-06	An instructor publishes lab slots (date, time, room, seats) for his or her own batches.	M4
FR-07	The system rejects a slot that overlaps another slot in the same room.	M4
FR-08	An instructor closes or cancels a slot; booked learners see the change.	M4
FR-09	A learner sees the open slots of his or her batch with seats left.	M5
FR-10	A learner books a seat; the system prevents overbooking and duplicate booking.	M5
FR-11	A learner cancels a booking before the slot starts.	M5
FR-12	An instructor marks present or absent against confirmed bookings; the slot then becomes COMPLETED.	M6
FR-13	The system computes course-wise attendance percentage per learner and flags values below the centre's threshold.	M6
FR-14	The administrator views and exports utilisation and attendance reports as CSV.	M6

Table 2.2: Non-functional requirements

Area	Requirement
Performance	Open-slot list and booking respond within [2] seconds for [50] concurrent users on the deployment hardware.
Concurrency	Two simultaneous requests for the last seat never produce two confirmed bookings.
Security	As in section 3.5: hashed passwords, HTTPS, role-based access, session timeout.
Usability	All learner screens usable on a phone screen 360 pixels wide.
Availability	Available during the centre's working hours [hours].
Maintainability	Layered code: controller, service and repository classes per module.
Data retention	No hard deletes of bookings or attendance; users are deactivated, not deleted.

2.3 Conceptual Models

The data flow diagrams follow the levelling approach described in [4]: a context diagram (Figure 2.1), a first-level diagram with six processes (Figure 2.2), and a second-level diagram for each first-level process (Figures 2.3 to 2.8), as Section III asks ("DFDs at least up to second level"). Duplicated external entities and data stores are marked with a short stroke. Figure 2.9 is the entity relationship diagram. Table 2.3 records the mapping that Annexure-2 asks for.

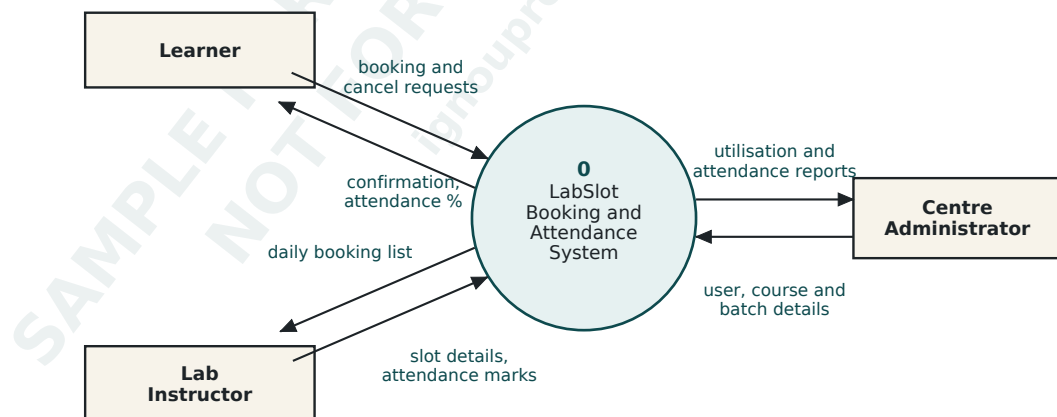


Figure 2.1: Context diagram (Level 0 DFD) of LabSlot

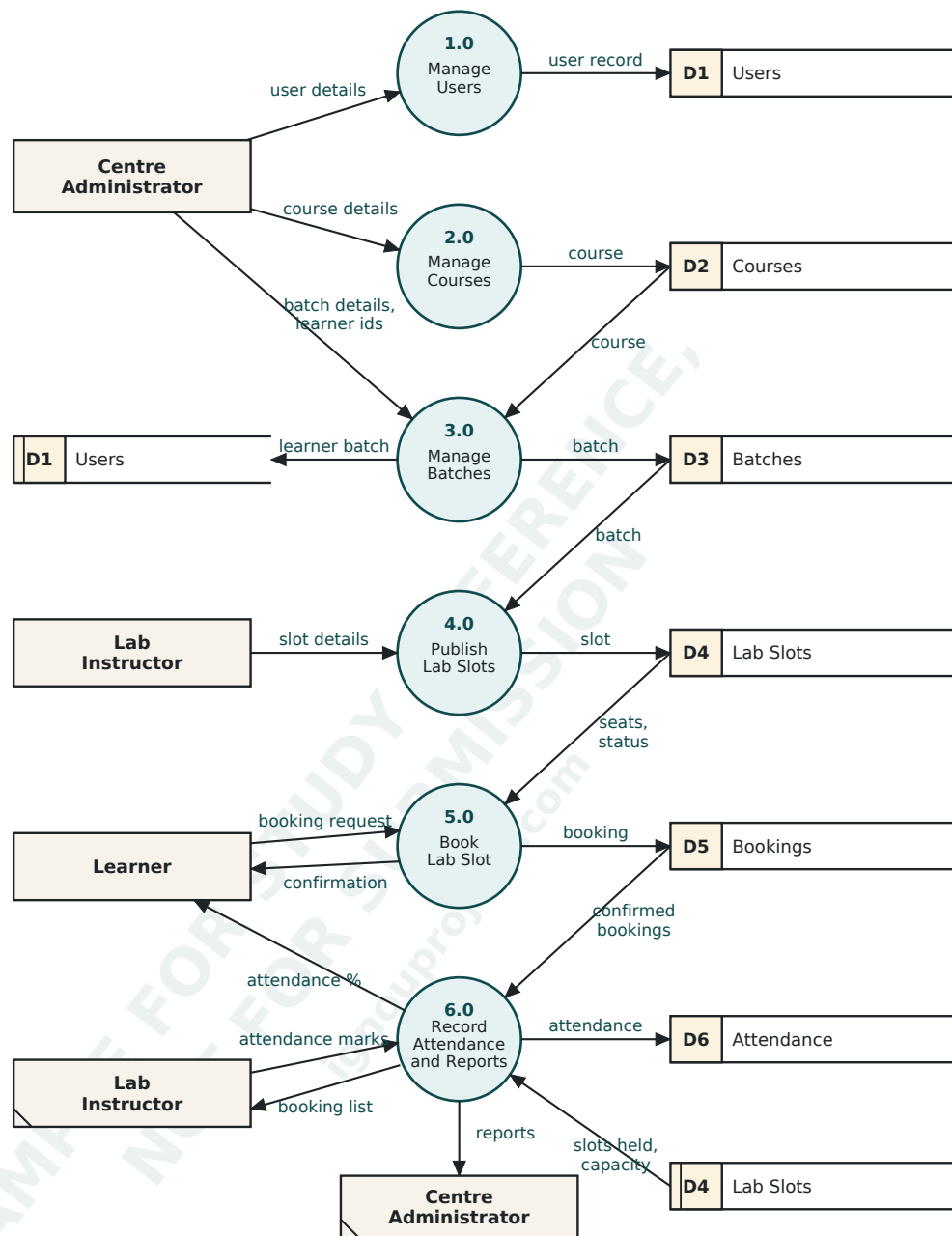


Figure 2.2: Level 1 DFD. All three roles log in through 1.0; login flows appear in Figure 2.3.

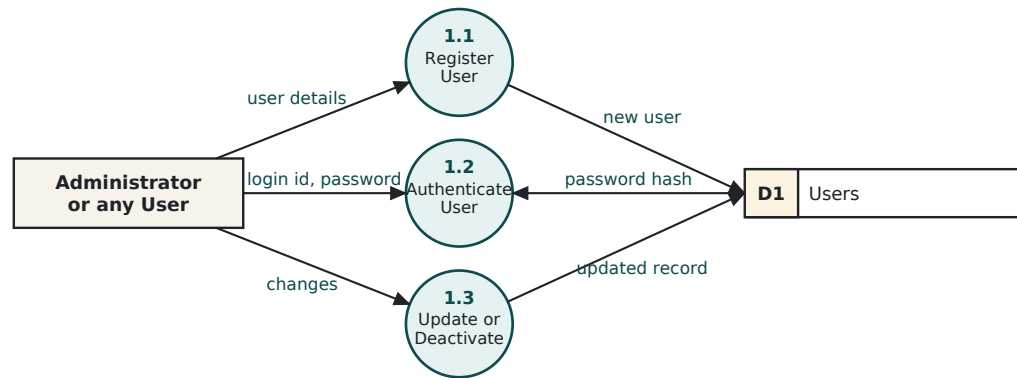


Figure 2.3: Level 2 DFD of process 1.0 Manage Users

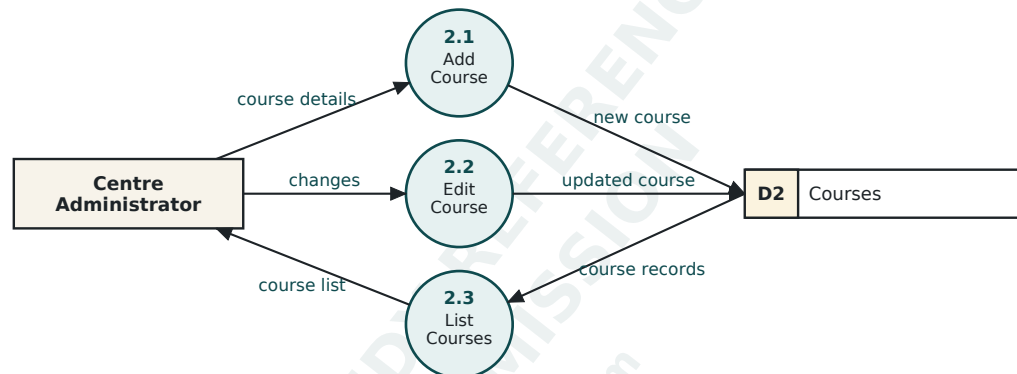


Figure 2.4: Level 2 DFD of process 2.0 Manage Courses

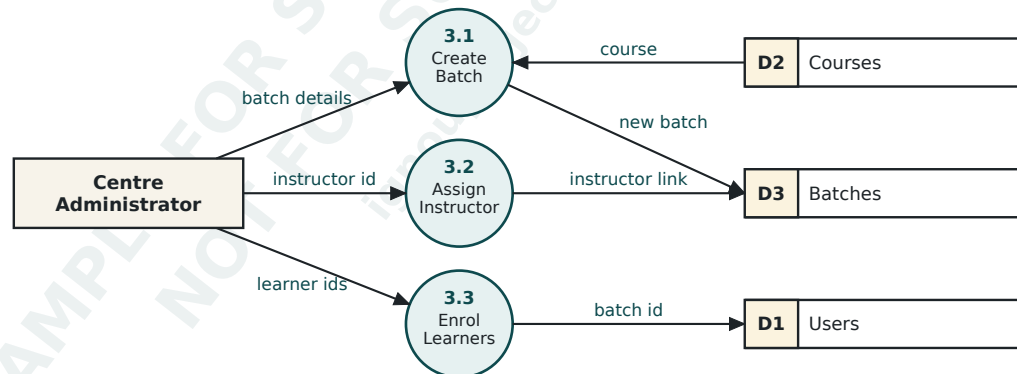


Figure 2.5: Level 2 DFD of process 3.0 Manage Batches

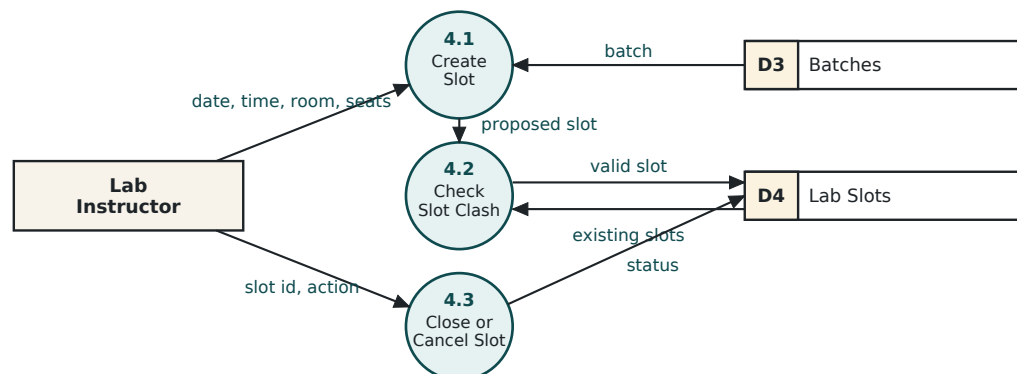


Figure 2.6: Level 2 DFD of process 4.0 Publish Lab Slots

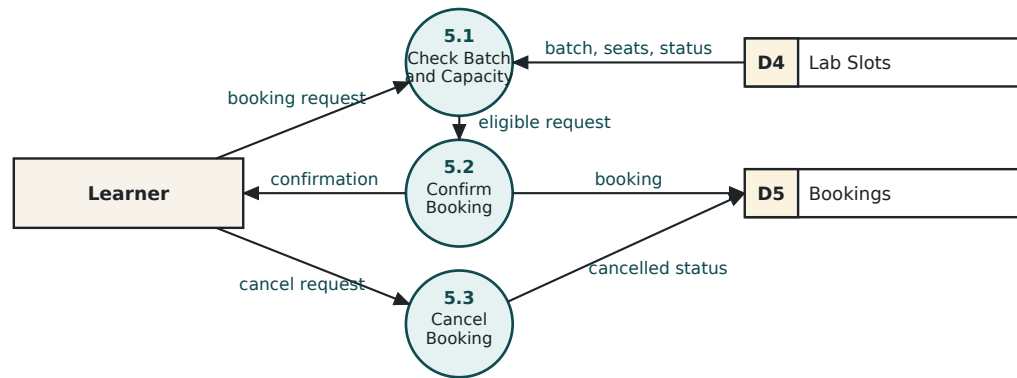


Figure 2.7: Level 2 DFD of process 5.0 Book Lab Slot

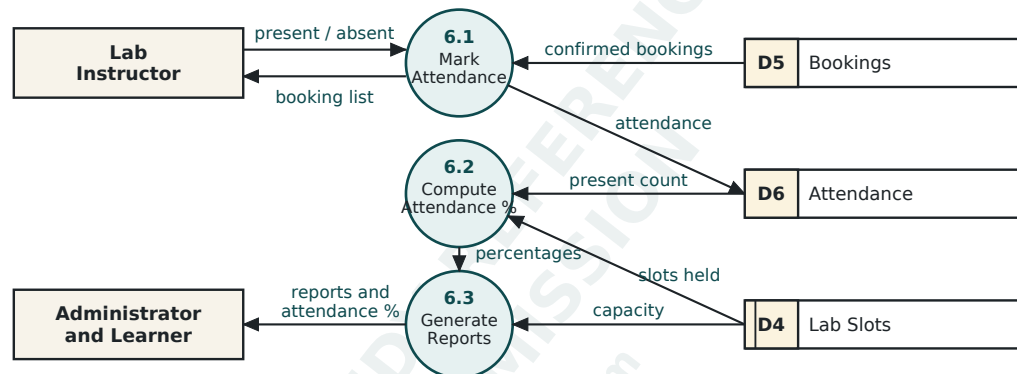


Figure 2.8: Level 2 DFD of process 6.0 Record Attendance and Reports

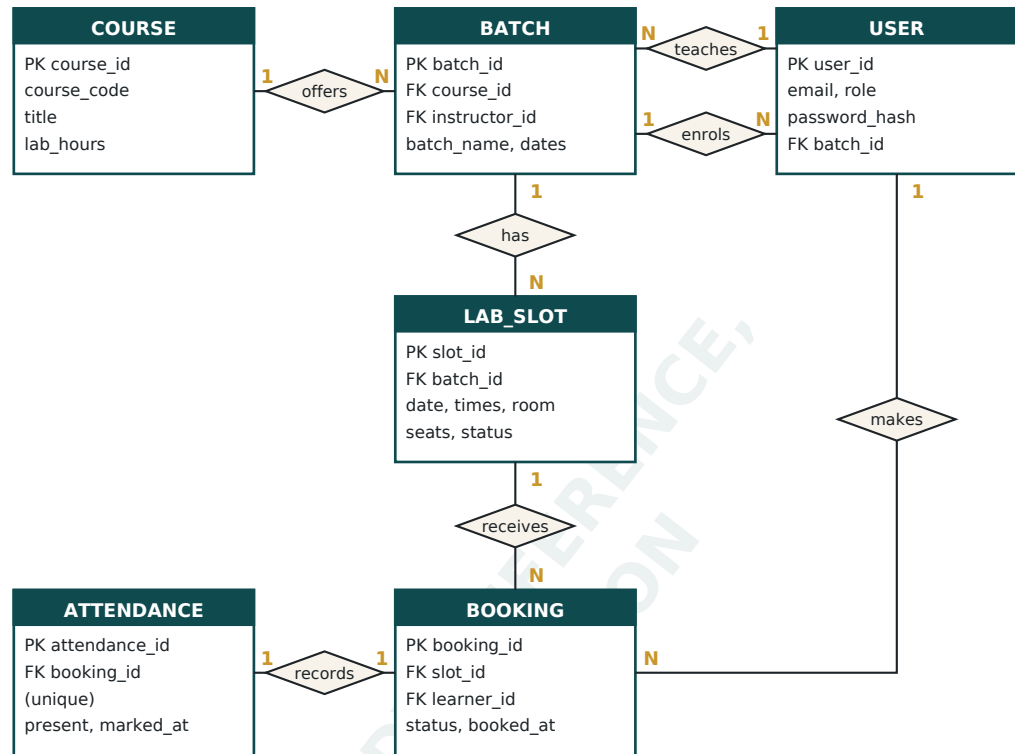


Figure 2.9: Entity relationship diagram. A USER is an administrator, instructor or learner (role attribute).

Table 2.3: Mapping required by Annexure-2 (modules, processes, data stores, entities, tables)

Module	Level 1 process	Level 2	Data store	Entity	Table
M1 User Management	1.0 Manage Users	1.1 to 1.3	D1 Users	USER	users
M2 Course Management	2.0 Manage Courses	2.1 to 2.3	D2 Courses	COURSE	courses
M3 Batch Management	3.0 Manage Batches	3.1 to 3.3	D3 Batches	BATCH	batches
M4 Lab Slot Management	4.0 Publish Lab Slots	4.1 to 4.3	D4 Lab Slots	LAB_SLOT	lab_slots
M5 Booking Management	5.0 Book Lab Slot	5.1 to 5.3	D5 Bookings	BOOKING	bookings
M6 Attendance and Reports	6.0 Record Attendance and Reports	6.1 to 6.3	D6 Attendance	ATTENDANCE	attendance

ignouproject.com

Sample synopsis for study reference only. Write your own; do not submit this file. |
WhatsApp +91 93546 37830

3. BASIC SYSTEM DESIGN

3.1 Basic Modules

Table 3.1 gives the six modules, the effort each represents and the main data structures each uses (Section III, item 5). Table 3.2 gives the tentative content of the reports.

Table 3.1: Modules, responsibilities and main data structures

Module	Responsibility	Main data structures
M1 User Management	Registration, login, password change, deactivation, account lock after [5] failed logins	User entity; Role enum (ADMIN, INSTRUCTOR, LEARNER); login form object
M2 Course Management	Add, edit and list courses	Course entity; list of courses for the selection menu
M3 Batch Management	Create batches, assign the instructor, enrol learners	Batch entity; list of learner ids selected for enrolment
M4 Lab Slot Management	Create slots, check room clashes, close or cancel slots	LabSlot entity; SlotStatus enum; time-interval comparison for clashes
M5 Booking Management	Show open slots, book with capacity check, cancel	Booking entity; BookingStatus enum; list of open slots with seats left
M6 Attendance and Reports	Attendance sheet from confirmed bookings, percentages, reports and CSV export	Attendance entity; map from learner id to present and held counts

Table 3.2: Reports generated (tentative content)

Report	User	Tentative content
R1 Daily booking list	Instructor	Slot, room, time, learners booked, seats left; used as the attendance sheet
R2 Slot utilisation	Administrator	Date range, slot, seats, confirmed bookings, present, utilisation percentage (present / seats)
R3 Attendance summary	Administrator; a learner sees his or her own row	Learner, batch, course, slots held, present, percentage, below-threshold flag
R4 Batch roster	Administrator, instructor	Batch, course, instructor, enrolled learners with contact e-mail

3.2 Database Design

3.2.1 Schema Design

The schema has six tables, one for each entity in Figure 2.9, normalised to third normal form [5]. The users table and the batches table refer to each other (a learner belongs to a batch; a batch has an instructor), so one of the two foreign keys is added after both tables exist.

Table 3.3: Structure of the six tables

Table	Column	Type	Key / constraint
users	user_id	INT, AUTO_INCREMENT	PRIMARY KEY
	full_name	VARCHAR(80)	NOT NULL
	email	VARCHAR(120)	NOT NULL, UNIQUE
	password_hash	CHAR(60)	NOT NULL (bcrypt output)
	role	ENUM('ADMIN', 'INSTRUCTOR', 'LEARNER')	NOT NULL
	batch_id	INT	NULL, FK to batches (learners only)
	active	TINYINT(1)	DEFAULT 1
	failed_logins	TINYINT	DEFAULT 0
courses	course_id	INT, AUTO_INCREMENT	PRIMARY KEY
	course_code	VARCHAR(20)	NOT NULL, UNIQUE
	title	VARCHAR(120)	NOT NULL
	lab_hours	SMALLINT	NOT NULL, CHECK (lab_hours above 0)
batches	batch_id	INT, AUTO_INCREMENT	PRIMARY KEY
	course_id	INT	NOT NULL, FK to courses
	instructor_id	INT	NOT NULL, FK to users
	batch_name	VARCHAR(40)	NOT NULL; UNIQUE with course_id
	start_date, end_date	DATE	start NOT NULL; end NULL
lab_slots	slot_id	INT, AUTO_INCREMENT	PRIMARY KEY
	batch_id	INT	NOT NULL, FK to batches
	slot_date	DATE	NOT NULL
	start_time, end_time	TIME	NOT NULL; CHECK end later than start
	lab_room	VARCHAR(20)	NOT NULL
	seats	SMALLINT	NOT NULL, CHECK (seats above 0)
	status	ENUM('OPEN', 'CLOSED', 'CANCELLED', 'COMPLETED')	DEFAULT 'OPEN'
bookings	booking_id	INT, AUTO_INCREMENT	PRIMARY KEY
	slot_id	INT	NOT NULL, FK to lab_slots
	learner_id	INT	NOT NULL, FK to users
	status	ENUM('CONFIRMED', 'CANCELLED')	NOT NULL
	booked_at	DATETIME	NOT NULL
	(slot_id, learner_id)		UNIQUE
attendance	attendance_id	INT, AUTO_INCREMENT	PRIMARY KEY
	booking_id	INT	NOT NULL, UNIQUE, FK to bookings
	present	TINYINT(1)	NOT NULL
	marked_at	DATETIME	NOT NULL

3.2.2 Data Integrity and Constraints

- **Entity integrity:** every table has an auto-increment primary key.
- **Referential integrity:** all foreign keys use ON DELETE RESTRICT, so a course, batch or slot with history cannot be deleted; users are deactivated instead.
- **Domain integrity:** ENUM columns limit roles and statuses; CHECK constraints keep seats and lab hours positive and end time later than start time. MySQL enforces CHECK constraints from version 8.0.16 [6].
- **Uniqueness:** one e-mail per user; one batch name per course; one booking row per learner per slot; one attendance row per booking.
- **Business rules enforced in the service layer:** a learner books only slots of his or her batch; confirmed bookings never exceed seats; two slots in the same room may not overlap on the same date; attendance is marked only for CONFIRMED bookings; booking closes when the slot starts.

3.3 Procedural Design

3.3.1 Flow Charts

Figure 3.1 shows the process logic of booking a seat (process 5.0), the step most likely to fail under simultaneous requests.

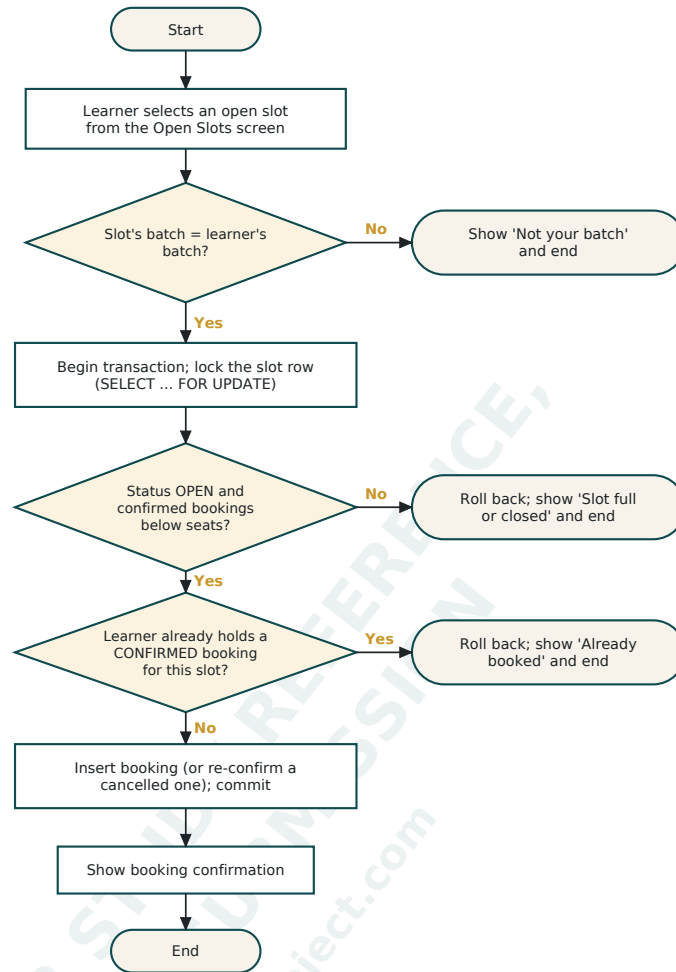


Figure 3.1: Flow chart of booking a lab seat

3.3.2 Algorithms

Algorithm 3.1 locks the slot row before counting bookings, so two requests for the last seat are served one after the other; InnoDB row locks taken by SELECT ... FOR UPDATE are released at commit or rollback [6]. Algorithm 3.2 computes attendance percentage over slots held for the learner's batch, so a learner who never books is not shown as 100 percent.

```

Algorithm 3.1 BookSlot(learnerId, slotId)
1 slot = find slot by slotId; if none: return "Slot not found"
2 if slot.batchId != batchOf(learnerId): return "Not your batch"
3 begin transaction
4 lock slot row (SELECT ... FOR UPDATE)
5 if slot.status != OPEN or slot has started:
    rollback; return "Slot closed"
6 booked = count bookings of slot with status CONFIRMED
7 if booked >= slot.seats: rollback; return "Slot full"
8 b = booking of (slotId, learnerId), if any
9 if b exists and b.status = CONFIRMED:
    rollback; return "Already booked"
10 if b exists: set b.status = CONFIRMED (re-booking)
    else: insert booking(slotId, learnerId, CONFIRMED, now)
11 commit; return "Booked"
  
```

```

Algorithm 3.2 AttendancePercent(learnerId)
1  batch = batchOf(learnerId)
2  held = count slots of batch with status COMPLETED
3  if held = 0: return "Not applicable"
4  present = count attendance rows with present = 1
      for bookings of learnerId in those slots
5  pct = round(present * 100 / held, 1)
6  flag = pct below centre threshold [for example 70]
7  return pct, flag

```

3.4 User Interface Design

Table 3.4: Planned screens

Screen	Role	Main elements
Login	All	E-mail, password, error message, lock notice
Dashboard	All	Role-specific menu; today's slots; counts
Users, Courses, Batches	Administrator	List with search; add and edit forms; enrol learners by multi-select
My Slots	Instructor	Calendar of own slots; create form; close or cancel buttons with confirmation
Attendance Sheet	Instructor	Booked learners of one slot with present or absent toggles; submit
Open Slots	Learner	Slots of own batch with seats left; Book button
My Bookings, My Attendance	Learner	Upcoming bookings with Cancel; percentage per course with flag
Reports	Administrator	Filters (date range, batch); R1 to R4 tables; CSV export

One Bootstrap layout with a fixed menu; field-level error messages; confirmation before any cancel or close action.

3.5 Security Issues

- Passwords are stored only as bcrypt hashes through Spring Security, an adaptive hashing method listed in [7].
- The whole application runs over HTTPS, so the user name and password are encrypted during transmission to the server, as Section V of the guidelines asks [1].
- Role-based access control on every URL and service method; a learner can read only his or her own bookings and attendance.
- Parameter binding through Spring Data JPA, so user input is never concatenated into SQL.
- CSRF tokens on all forms, session timeout after [15] minutes of inactivity, and account lock after [5] failed logins.
- Audit columns (booked_at, marked_at) and a daily database backup [schedule to be agreed with the centre].

3.6 Test Cases Design

Testing process: unit tests with JUnit 5 and Mockito for service classes; integration tests with Spring Boot Test against a separate MySQL test schema; system tests in a browser against Table 2.1. Table 3.5 gives the planned cases and their expected results. Actual results will be recorded only in the project report, after the tests are run.

Table 3.5: Planned test cases (design only, no results)

TC	Level	Condition or input	Expected result
01	Unit, M1	Register with an e-mail already in use	Rejected: e-mail already registered
02	Unit, M1	Wrong password [5] times	Account locked for [15] minutes
03	Unit, M2	Duplicate course code	Rejected
04	Unit, M4	End time earlier than start time	Rejected by validation and CHECK constraint
05	Unit, M4	Same room, same date, overlapping time	Rejected as a clash
06	Unit, M5	Learner books a slot of another batch	Rejected: Not your batch
07	Integration, M4 and M5	Booking when confirmed bookings equal seats	Rejected: Slot full
08	Integration, M5	Two simultaneous requests for the last seat	Exactly one CONFIRMED booking
09	Unit, M5	Re-book after cancelling	Same row set to CONFIRMED; no duplicate row
10	Integration, M5 and M6	Mark attendance for a learner without a confirmed booking	Not allowed
11	Unit, M6	Attendance percentage when no slot is completed	Not applicable
12	System	Learner opens another learner's attendance URL	Access denied (HTTP 403)
13	System	Login page requested over HTTP	Redirected to HTTPS

3.7 Future Scope and Further Enhancement

Section III, item 8 asks for future scope; Annexure-2 has no heading for it, so it is added here. Planned enhancements: a waiting list that offers a freed seat to the next learner; reminders through an SMS or messaging gateway; QR-code check-in at the lab door; RFID or biometric attendance devices; a mobile application; and support for several centres from one installation.

ignouproject.com

Sample synopsis for study reference only. Write your own; do not submit this file. |
WhatsApp +91 93546 37830

REFERENCES

- [1] IGNOU School of Computer and Information Sciences, 2026, *Project Guidelines: BCSP-165*, Indira Gandhi National Open University, New Delhi.
- [2] Spring, *Spring Boot Reference Documentation*, <https://docs.spring.io/spring-boot/>, last accessed [date].
- [3] IEEE, 1998, *IEEE Recommended Practice for Software Requirements Specifications*, IEEE Std 830-1998, IEEE, New York.
- [4] R. S. Pressman and B. R. Maxim, 2020, *Software Engineering: A Practitioner's Approach*, 9th edition, McGraw-Hill Education, New York.
- [5] R. Elmasri and S. B. Navathe, 2016, *Fundamentals of Database Systems*, 7th edition, Pearson, Boston.
- [6] Oracle, *MySQL 8.0 Reference Manual*, <https://dev.mysql.com/doc/refman/8.0/en/>, last accessed [date].
- [7] OWASP Foundation, *Password Storage Cheat Sheet*, https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html, last accessed [date].

References are numbered in the order of first use, as Section 2 of the guidelines asks. Check every web address and add the date you last opened it.

ignouproject.com

Sample synopsis for study reference only. Need your own draft checked?
ignouproject.com/ignou-project-synopsis-correction/ | WhatsApp +91 93546 37830