

Session 3

Software Requirements Specification for the Railway Reservation System

This session produces the Software Requirements Specification (SRS) for the Railway Reservation System (RRS) whose scope you wrote in Session 1 and whose size you estimated in Session 2. The SRS is the contract between the people who want the system and the people who will build it. Every design decision in Sessions 5 and 6, every test case in Session 8, and the program in Session 13 must point back to a numbered requirement on this page. If a requirement is missing or vague here, the mistake travels through every later session and costs the most to fix.

Objectives

✕ Do not copy. Read for understanding and the viva

- Write an SRS in the IEEE 830 shape that the manual (section 1.3.2) asks for.
- State each functional requirement as one numbered, testable “shall” sentence.
- Cover all eight RRS modules and every business rule of the case study.
- Write non-functional requirements with numbers, not adjectives.
- Build a traceability table so later sessions can cite requirement IDs.

Problem Statement

■ Write in lab record

Develop SRS (Software Requirements Specification) for the Railway Reservation System (RRS).

Concept

✕ Do not copy. Read for understanding and the viva

What an SRS is for

The scope statement (Session 1) says what the product is. The SRS says exactly what it must do, from the outside, without saying how. Designers read it to build the system. Testers read it to write test cases. The customer reads it to sign off. So it must be complete (nothing important left out), consistent (no two requirements contradict), unambiguous (one reading only), and verifiable (you can run a test that passes or fails).

The IEEE 830 shape

The manual gives the IEEE format: Introduction, Overall Description, Specific Requirements, Appendices. Section 1 tells the reader why the document exists and how to read it. Section 2 gives the background that makes the requirements understandable: who the users are, what the product sits next to, what limits apply. Section 3 is the real content: every requirement, numbered. Nothing in sections 1 and 2 is a requirement; if a sentence there says "shall", move it to section 3.

Writing one requirement

A good requirement has the pattern: `The system shall <verb> <object> <condition>`. One requirement per sentence. Use "shall" for mandatory, "should" for desirable. Avoid "user friendly", "fast", "secure", "etc." because no test can check them. Compare:

Weak	Better
The system should be fast.	The system shall return search results within 2 seconds for 95 percent of requests at 500 concurrent users.
Users can cancel tickets.	The system shall allow the booking owner or a Reservation Clerk to cancel a booking by PNR before the train departs from the source station.

Functional versus non-functional

Functional requirements (FR) describe behaviour: input, processing, output. Non-functional requirements (NFR) describe qualities of the whole system: speed, security, uptime, ease of use, ease of change. Both get IDs. Here the FR ID is `FR-<module>.<number>`, so `FR-4.3` is the

third requirement of module 4 (Booking). NFRs are `NFR-<letter><number>`, for example `NFR-P1` for performance.

Traceability

Traceability means every requirement can be followed forward to the design element, code, and test that satisfy it, and backward to the scope item or user need it came from. The table at the end of the SRS is the start of that chain. In Session 4 the DFD processes are numbered 1.0 to 8.0 to match the modules, so `FR-4.x` maps to process 4.0.

Software Requirements Specification

■ Write in lab record

Software Requirements Specification for the Railway Reservation System (RRS) Version 1.0, prepared for MCS-217 Session 3.

1 Introduction

1.1 Purpose

This SRS states the functional and non-functional requirements of the Railway Reservation System (RRS). It is written for the designers who will produce the modular design (Session 5) and user interface design (Session 6), the testers who will derive test cases, and the developers who will implement the system (Session 13). It also serves as the agreed statement of what the customer will receive.

1.2 Scope

The RRS is a web-based system through which passengers search trains, check seat availability, book and cancel tickets, and pay online; reservation clerks book and cancel at counters; and administrators maintain trains, routes, coaches, fares, and schedules and view reports. The system generates a unique PNR for every booking, maintains a waitlist per schedule and class, applies time-based refund rules on cancellation, and sends SMS and email notifications through an external service. Out of scope: Tatkal quota, concession categories, catering, parcel booking, and integration with the national railway network.

1.3 Definitions, acronyms and abbreviations

Term	Meaning
RRS	Railway Reservation System, the software described here
PNR	Passenger Name Record, a 10-digit unique number that identifies one booking
Schedule	One run of a train on one date
Route	The ordered list of stations a train passes, with times and distances
Class	Coach class: SL (Sleeper), 3A (AC 3-tier), 2A (AC 2-tier), 1A (AC First)
Berth	A seat in a coach; types LOWER, MIDDLE, UPPER, SIDE LOWER, SIDE UPPER
Waitlist (WL)	Queue of bookings waiting for a seat in a class of a schedule
Clerkage	Flat charge deducted from a refund
Gateway	External payment gateway used for online payment
SRS	Software Requirements Specification
FR, NFR	Functional requirement, non-functional requirement

1.4 References

- IEEE Std 830-1998, Recommended Practice for Software Requirements Specifications.
- Statement of Scope for RRS, [Session 1](#).
- Effort and cost estimate for RRS, [Session 2](#).
- MCS-213 Software Engineering, Block 1 (Requirements Analysis).

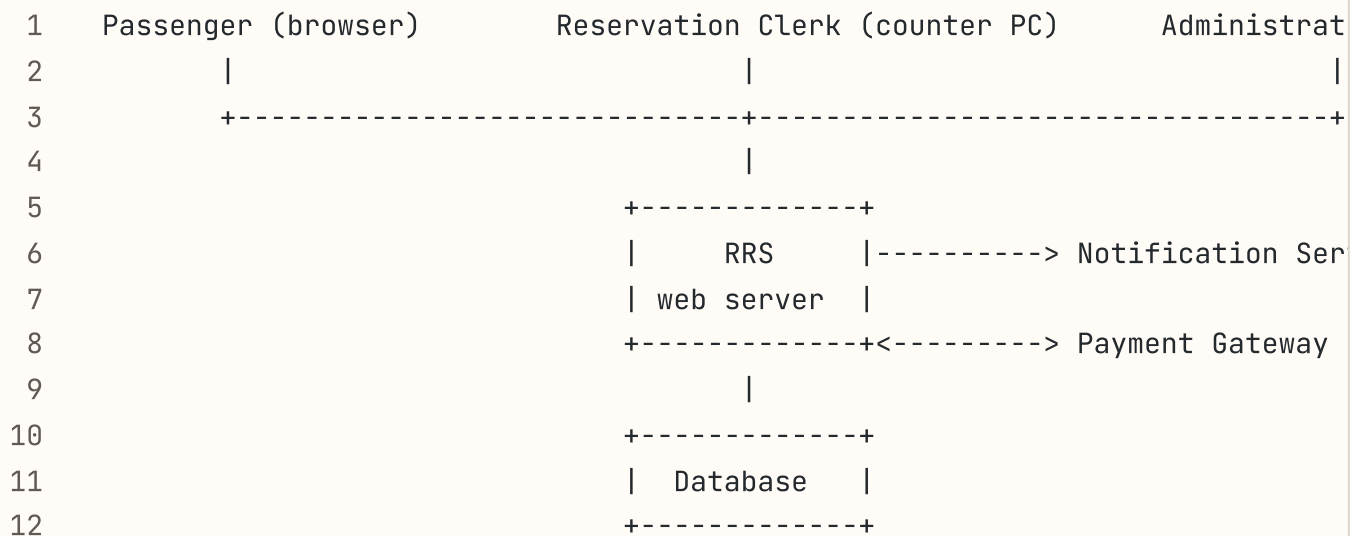
1.5 Overview

Section 2 describes the product, its users, environment, constraints and assumptions. Section 3 lists the specific requirements: functional requirements by module, external interfaces, and non-functional requirements. Section 4 gives the traceability table.

2 Overall Description

2.1 Product perspective

The RRS is a new, self-contained system. It replaces manual counter registers and a spreadsheet-based seat chart. It depends on two external systems: a Payment Gateway for online payment and a Notification Service for SMS and email. It stores all data in a single relational database shared by the web application and the counter application.



2.2 Product functions

The eight modules and their main functions:

No	Module	Main functions
1	User Management	Register, login, logout, role control, profile update
2	Train and Schedule Management	Maintain stations, trains, routes, coaches, fares, schedules
3	Search and Availability	Search trains by source, destination, date; show seats and fare per class
4	Booking	Capture passengers, allocate seats or waitlist, generate PNR, PNR enquiry
5	Cancellation and Refund	Cancel by PNR, compute refund, promote waitlist
6	Payment	Initiate, confirm, fail, refund through the gateway; cash at counter
7	Notification	Send booking, cancellation, waitlist and schedule-change messages
8	Reports	Occupancy, revenue, cancellation reports for administrators

2.3 User classes and characteristics

User class	Description	Frequency of use	Technical skill
Passenger	Public user who books for self or others online	Occasional	Low; may use a phone browser
Reservation Clerk	Railway staff at a counter; books and cancels on behalf of walk-in passengers, takes cash	Continuous during counter hours	Trained on the system
Administrator	Railway operations staff; maintains master data and schedules, views reports	Daily	Trained; understands railway operations
Payment Gateway	External system; receives payment requests, returns result	Every online payment	Not applicable
Notification Service	External system; sends SMS and email	Every booking event	Not applicable

2.4 Operating environment

- Server: Linux, Python 3 web application, PostgreSQL 14 or later database.
- Client: any browser released in the last three years, desktop or mobile, no plug-ins.
- Counter PC: same browser client on the railway intranet, with a receipt printer.
- Network: HTTPS over the public internet for passengers; intranet for clerks and administrators.

2.5 Design and implementation constraints

- Card numbers and CVV shall never be stored; all card handling happens on the gateway page.
- All timestamps are stored in Indian Standard Time.
- The database is the single source of truth for seat allocation; no seat map is cached on the client.
- The system must run on the hardware already available at counters (4 GB RAM PCs).
- Development effort is bounded by the Session 2 estimate (225 adjusted FP, about 13,500 LOC).

2.6 Assumptions and dependencies

- The Payment Gateway responds within 5 minutes to every payment request, with a final SUCCESS or FAILURE.
- The Notification Service accepts messages through an HTTP API and reports delivery status.
- Station codes, train numbers, and routes are supplied by the administrator from official railway data.
- One train runs at most once per day, so (train_no, run_date) identifies a schedule.
- Fares depend only on train, class, and distance; no dynamic pricing.

3 Specific Requirements

3.1 Functional requirements

Priority: M = must have, S = should have.

Module 1: User Management

ID	Requirement	Priority
FR-1.1	The system shall allow a new user to register with name, email, mobile number, and password. Email and mobile shall each be unique across users.	M
FR-1.2	The system shall store passwords only as salted hashes and shall never display or email a password.	M
FR-1.3	The system shall authenticate a user by email and password and shall lock the account for 15 minutes after 5 consecutive failed attempts.	M
FR-1.4	Every user shall have exactly one role: PASSENGER, CLERK, or ADMIN. Only an ADMIN shall be able to create CLERK and ADMIN accounts; self-registration always creates a PASSENGER.	M
FR-1.5	The system shall allow a logged-in user to change mobile number and password after re-entering the current password.	S
FR-1.6	The system shall end a session after 20 minutes of inactivity and shall require login again.	M

Module 2: Train and Schedule Management

ID	Requirement	Priority
FR-2.1	The system shall allow an ADMIN to add and edit stations with station_code (2 to 5 upper-case letters, unique), name, and city.	M
FR-2.2	The system shall allow an ADMIN to add a train with train_no (5 digits, unique), name, type (EXPRESS, SUPERFAST, PASSENGER), and total_coaches.	M
FR-2.3	The system shall allow an ADMIN to define a route for a train as an ordered list of stations with arrival time, departure time, and cumulative distance in km. Sequence numbers shall start at 1 and increase by 1 with no gaps.	M
FR-2.4	The system shall allow an ADMIN to add coaches to a train with class (SL, 3A, 2A, 1A) and seat_count, and shall generate one Seat record per seat with seat_no and berth_type automatically.	M
FR-2.5	The system shall allow an ADMIN to set a fare rate_per_km for every (train, class) pair. A train shall not be schedulable until every class of its coaches has a fare.	M
FR-2.6	The system shall allow an ADMIN to create a schedule (train_no, run_date) with status SCHEDULED, and to change the status to CANCELLED or RESCHEDULED.	M
FR-2.7	The system shall reject a schedule whose run_date is earlier than the current date, and shall reject a duplicate (train_no, run_date).	M

Module 3: Search and Availability

ID	Requirement	Priority
FR-3.1	The system shall let any user, without login, search trains by source station, destination station, and travel date.	M
FR-3.2	The search shall return only trains whose route contains the source before the destination and which have a schedule on the given date with status SCHEDULED.	M
FR-3.3	For each train found, the system shall show train_no, name, departure time at source, arrival time at destination, distance, and journey duration, sorted by departure time.	M
FR-3.4	For each class of each train found, the system shall show the number of available seats, or "WL n" where n is the number of passengers already waitlisted in that class.	M
FR-3.5	For each class, the system shall show the fare per passenger, computed as rate_per_km multiplied by the distance between source and destination, rounded to the nearest rupee.	M
FR-3.6	The system shall refresh availability from the database at the time of each search; it shall not show cached counts older than 30 seconds.	S

Module 4: Booking

ID	Requirement	Priority
FR-4.1	The system shall allow only a logged-in PASSENGER or CLERK to start a booking.	M
FR-4.2	The system shall accept between 1 and 6 passengers per booking, each with name, age, gender, and berth_preference. A seventh passenger shall be rejected with a message.	M
FR-4.3	The system shall refuse a booking if the schedule status is not SCHEDULED or if the departure time at the source station has already passed.	M
FR-4.4	The system shall compute total_fare as the per-passenger fare (FR-3.5) multiplied by the number of passengers.	M
FR-4.5	When enough seats are free in the requested class, the system shall allocate one seat per passenger, honouring berth_preference where possible and keeping all passengers of one booking in the same coach where possible.	M
FR-4.6	When seats are not enough for all passengers, the system shall place the whole booking on the waitlist for that schedule and class with the next waitlist number. A booking shall never be split between confirmed and waitlisted passengers.	M
FR-4.7	The system shall hold allocated seats for 10 minutes while payment is pending and shall release them if payment is not successful within that time.	M
FR-4.8	On successful payment the system shall generate a unique 10-digit PNR, set the booking status to CONFIRMED or WAITLISTED, and record booking_time.	M
FR-4.9	The system shall let any user, without login, enter a PNR and see the train, date, passengers, allocated seats or waitlist numbers, and booking status.	M

Module 5: Cancellation and Refund

ID	Requirement	Priority
FR-5.1	The system shall allow the booking owner or a CLERK to cancel a booking by PNR, either all passengers or selected passengers.	M
FR-5.2	The system shall refuse cancellation after the departure time of the train at the source station.	M
FR-5.3	The system shall compute the refund per cancelled passenger from the hours between cancellation and departure at source: more than 48 hours, 100 percent of the fare minus a flat clerkage of 60 rupees; 48 hours down to 12 hours, 50 percent of the fare; less than 12 hours, no refund.	M
FR-5.4	When all passengers of a booking are cancelled the system shall set the booking status to CANCELLED; when some are cancelled the booking stays in its current status with the remaining passengers.	M
FR-5.5	When confirmed seats are released, the system shall promote waitlisted bookings of the same schedule and class in waitlist order, one booking at a time, only when the freed seats are enough for all passengers of that booking.	M
FR-5.6	For every cancellation with a non-zero refund the system shall create a Refund record and pass it to the Payment module for return to the original payment mode.	M
FR-5.7	The system shall automatically cancel every booking still WAITLISTED at departure time and refund the full fare without clerkage.	S

Module 6: Payment

ID	Requirement	Priority
FR-6.1	The system shall create a Payment record with amount equal to total_fare, mode (UPI, CARD, NETBANKING, CASH), and status INITIATED before contacting the gateway.	M
FR-6.2	For online modes the system shall send the amount and a unique order reference to the Payment Gateway and shall store the txn_ref the gateway returns.	M
FR-6.3	On a SUCCESS response the system shall set the payment status to SUCCESS and paid_at to the current time, then complete the booking (FR-4.8).	M
FR-6.4	On a FAILURE response, or no response within 5 minutes, the system shall set the payment status to FAILED and release the held seats (FR-4.7).	M
FR-6.5	For CLERK bookings the system shall accept mode CASH, skip the gateway, and mark the payment SUCCESS on the clerk's confirmation.	M
FR-6.6	The system shall send refund instructions to the gateway for online payments and shall set the refund status to PROCESSED when the gateway confirms; cash refunds are marked PROCESSED by the clerk.	M
FR-6.7	The system shall never store card numbers, expiry dates, or CVV.	M

Module 7: Notification

ID	Requirement	Priority
FR-7.1	On booking completion the system shall send an SMS and an email to the booking user with PNR, train, date, source, destination, passengers with seats or waitlist numbers, and total fare.	M
FR-7.2	On cancellation the system shall send an SMS and an email with PNR, cancelled passengers, and refund amount.	M
FR-7.3	On promotion from waitlist the system shall notify the booking user with the new seat numbers.	M
FR-7.4	When a schedule is set to CANCELLED or RESCHEDULED the system shall notify every CONFIRMED and WAITLISTED booking on that schedule within 15 minutes.	M
FR-7.5	The system shall log every notification with time, channel, and delivery status, and shall retry a failed send up to 3 times at 5-minute intervals.	S

Module 8: Reports

ID	Requirement	Priority
FR-8.1	The system shall allow only an ADMIN to open the reports module.	M
FR-8.2	The system shall produce an occupancy report for a schedule showing, per class, total seats, confirmed, available, waitlisted, and percentage occupancy.	M
FR-8.3	The system shall produce a revenue report for a date range showing, per train and class, number of bookings, gross fare, refunds paid, and net revenue.	M
FR-8.4	The system shall produce a cancellation report for a date range showing count and refund total in each of the three refund time bands.	M
FR-8.5	The system shall allow every report to be downloaded as a CSV file.	S

3.2 External interface requirements

User interfaces

Screen	Users	Content
Register, Login	All	Form fields with inline validation messages next to the field in error
Search	All	Source, destination, date; results table with class-wise availability and fare
Booking form	Passenger, Clerk	Up to 6 passenger rows, class, fare summary, proceed to pay
PNR status	All	PNR input; passenger list with seat or WL number and status
Cancellation	Passenger, Clerk	PNR lookup, passenger tick boxes, refund preview before confirm
Admin console	Admin	Tabs for stations, trains, routes, coaches, fares, schedules, reports

Every screen shall work at 360 px width (phone) and 1280 px width (desktop). Every error shall say what was wrong and what to do.

Hardware interfaces

- Receipt printer at counters through the browser print dialog; the ticket page shall have a print layout on A5 paper.

Software interfaces

Interface	Direction	Protocol and data
Database	RRS to PostgreSQL	SQL over TCP; tables per the data dictionary in Session 4
Payment Gateway	RRS to gateway and back	HTTPS REST; request: order reference, amount, currency, return URL; response: txn_ref, status
Notification Service	RRS to service	HTTPS REST; request: channel (SMS or EMAIL), recipient, message; response: message id, status

Communication interfaces

- All browser traffic over HTTPS with TLS 1.2 or later.
- Gateway and notification calls authenticated with API keys kept in server configuration, not in source code.

3.3 Non-functional requirements

ID	Category	Requirement
NFR-P1	Performance	Search results shall appear within 2 seconds for 95 percent of requests at 500 concurrent users.
NFR-P2	Performance	A booking shall be confirmed and its PNR shown within 5 seconds after the gateway returns SUCCESS.
NFR-P3	Performance	The system shall support 1,000 logged-in users and 100 bookings per minute at peak.
NFR-S1	Security	All pages shall be served only over HTTPS; HTTP requests shall be redirected.
NFR-S2	Security	Passwords shall be hashed with bcrypt (cost 12 or higher).
NFR-S3	Security	Role checks shall be enforced on the server for every request; hiding a menu item is not sufficient.
NFR-S4	Security	Every booking, cancellation, refund, and admin change shall be written to an audit log with user id and timestamp, retained for 1 year.
NFR-R1	Reliability	A seat on a schedule shall never be allocated to two passengers; concurrent bookings shall be serialised by database transactions.
NFR-R2	Reliability	A failed payment shall never leave a booking in CONFIRMED status; a successful payment shall never leave a booking without a PNR.
NFR-R3	Reliability	The database shall be backed up every 6 hours and a restore shall lose at most 6 hours of data.
NFR-A1	Availability	The system shall be available 99.5 percent of each calendar month, measured outside the maintenance window.
NFR-A2	Availability	Planned maintenance shall occur only between 02:00 and 04:00 IST and shall be announced on the login page 24 hours earlier.

ID	Category	Requirement
NFR-U1	Usability	A first-time passenger shall complete a search and booking in under 5 minutes without help, measured with 5 test users.
NFR-U2	Usability	Every input error shall be shown next to the field, in plain language, without clearing the other fields.
NFR-U3	Usability	The interface shall be in English and Hindi, selectable on every page.
NFR-M1	Maintainability	Each of the eight modules shall be a separate package with no direct access to another module's tables; cross-module calls go through documented functions.
NFR-M2	Maintainability	Adding a new coach class or changing the clerkage amount shall require a configuration change only, no code change.
NFR-M3	Maintainability	Unit tests shall cover at least 70 percent of statements in modules 4, 5, and 6.

4 Requirements traceability

Module	FR IDs	Scope item (Session 1)	DFD process (Session 4)	Main data stores
1 User Management	FR-1.1 to FR-1.6	Login facility, secure access	1.0 Manage Users	D1 User
2 Train and Schedule Management	FR-2.1 to FR-2.7	Update train information and schedule	2.0 Manage Trains and Schedules	D2 Train Master, D3 Schedule
3 Search and Availability	FR-3.1 to FR-3.6	Seat availability, journey time	3.0 Search Trains	D2 Train Master, D3 Schedule, D4 Booking
4 Booking	FR-4.1 to FR-4.9	Create reservation, view reservation status	4.0 Book Ticket	D4 Booking, D2 Train Master
5 Cancellation and Refund	FR-5.1 to FR-5.7	Cancel reservation, easy and fast refund	5.0 Cancel and Refund	D4 Booking, D6 Refund
6 Payment	FR-6.1 to FR-6.7	All payment modes supported	6.0 Process Payment	D5 Payment
7 Notification	FR-7.1 to FR-7.5	Details sent to email and mobile	7.0 Send Notifications	D7 Notification Log
8 Reports	FR-8.1 to FR-8.5	Generating reports	8.0 Generate Reports	D3, D4, D5, D6

Non-functional requirements apply across modules: NFR-S1 to NFR-S4 trace to the security paragraph of the scope; NFR-A1 and NFR-A2 to “twenty-four-7 accessibility”; NFR-U1 to NFR-U3 to “user friendly, simple interface” and “multilingual”.

Viva Questions

× Do not copy. Read for understanding and the viva

- **Q:** What is the difference between the scope statement and the SRS? **A:** The scope says what the product is and its boundary; the SRS states every requirement precisely enough to design and test against.
- **Q:** Why must a requirement be verifiable? **A:** Because a tester needs to write a test that clearly passes or fails; “the system should be fast” cannot be tested.
- **Q:** What does “shall” mean in an SRS? **A:** A mandatory requirement; “should” marks a desirable one.
- **Q:** Where in the IEEE 830 format do functional requirements go? **A:** Section 3, Specific Requirements; sections 1 and 2 give context only.
- **Q:** Give one non-functional requirement of the RRS with a number in it. **A:** NFR-P1: search results within 2 seconds for 95 percent of requests at 500 concurrent users.
- **Q:** What is traceability? **A:** The ability to follow a requirement forward to design, code, and tests, and backward to the need that produced it.
- **Q:** Why is the refund rule in the SRS and not left to the design? **A:** It is a business rule the customer must agree to; the design only decides how to compute it.
- **Q:** What are the external interfaces of the RRS? **A:** Browser user interface, receipt printer, database, Payment Gateway API, Notification Service API.
- **Q:** What happens if a payment fails after seats are held? **A:** FR-6.4 and FR-4.7: the payment is marked FAILED and the held seats are released after the 10-minute hold.

Common Mistakes

× Do not copy. Read for understanding and the viva

- Writing the SRS as a feature list of one-line bullets without IDs; then no later session can refer to a requirement.
- Mixing design into requirements (“the system shall use a hash table for seats”). Say what, not how.

- Non-functional requirements without numbers: "secure", "reliable", "user friendly" are wishes, not requirements.
- Leaving out the business rules (refund bands, max 6 passengers, no booking after departure). These are the requirements a viva examiner asks about first.
- Forgetting the external actors. Payment Gateway and Notification Service are not users, but their interfaces must be specified.
- Two requirements that contradict each other, for example one saying search needs login and another saying it does not. Read the whole list once for consistency before submitting.

Session Summary

■ Write in lab record

- Problem statement copied verbatim.
- SRS sections 1 and 2: introduction, definitions table, product perspective diagram, product functions table, user classes table, environment, constraints, assumptions.
- Section 3.1: functional requirements tables for all eight modules, FR-1.1 to FR-8.5, each a single testable "shall" sentence with priority.
- Section 3.2: user, hardware, software, and communication interfaces.
- Section 3.3: non-functional requirements NFR-P, S, R, A, U, M, each with a measurable value.
- Section 4: traceability table from modules and FR ranges to scope items and DFD processes.

[✎ Edit this page](#)

< Previous
Session 2

Next >
Session 4