

Session 1

Statement of Scope for the Railway Reservation System

This session produces the first document of the Railway Reservation System (RRS) project: the statement of scope. Every later session (SRS, DFDs, ERD, design, estimation, code, change control) builds on the boundaries you fix here. A scope statement says what the software will do, what it will not do, who uses it, and under what limits it must work. Getting it wrong means every later document argues with itself, so write it carefully and keep it short enough to defend in a viva.

Objectives

✕ Do not copy. Read for understanding and the viva

- Explain what a statement of scope is and where it sits in a project plan.
- Identify the actors, modules, and boundaries of the Railway Reservation System.
- Write a complete scope statement with purpose, in-scope and out-of-scope items, constraints, assumptions, deliverables, and success criteria.
- Use the same names for actors, modules, and entities that the rest of the lab record will use.

Problem Statement

■ Write in lab record

Session 1: Suppose that you need to build software for a Railway Reservation System. Write a statement of scope that describes the software.

Concept

✕ Do not copy. Read for understanding and the viva

What a scope statement is

A statement of scope is the first section of a project plan. The MCS-217 manual places it under “Overall Goal of the Project”, before objectives, deliverables, milestones, and budget. It answers four questions in plain language: what problem the software solves, what functions it provides, who will use it, and what is excluded. It is written before the SRS, so it names functions but does not detail inputs, outputs, and screens. That detail comes in Session 3.

Why boundaries matter more than features

Students usually write a long wish list (maps, voice input, multilingual menus) and call it scope. A scope statement is more useful for what it leaves out. Every excluded item is a decision the project team no longer has to estimate, design, code, or test. The out-of-scope list is what the examiner reads to see whether you understood the size of the job.

Actors and modules

An actor is a person or an external system that exchanges data with the software. The RRS has three human actors (Passenger, Reservation Clerk, Administrator) and two external systems (Payment Gateway, Notification Service). A module is a group of related functions that one actor mainly uses. The RRS is divided into eight modules, listed in the scope table below. Session 2 counts function points per module, Session 5 designs one component per module, and Session 13 implements four of them, so the module list must not change after this session.

Constraints and assumptions

A constraint is a limit the team cannot change: the platform, a law, a performance target. An assumption is something the team takes as true without proof, such as “the railway supplies the station list”. Write both explicitly. If an assumption turns out false, the scope changes, and the change goes through Session 14’s change control process.

Statement of Scope

■ Write in lab record

1. Project title

Railway Reservation System (RRS), version 1.0.

2. Purpose

The RRS is a web-based software system that lets passengers search trains, check seat availability, book and cancel reserved tickets, and pay online, and lets railway staff manage trains, schedules, fares, and reports. It replaces manual counter booking registers with a single shared database so that a seat booked at any counter or online is immediately unavailable everywhere else. The system reduces passenger time spent at reservation counters, removes double booking, and gives the administration daily occupancy and revenue figures without manual tallying.

3. Actors

Actor	Type	What the actor does in the RRS
Passenger	Human, external user	Registers, logs in, searches trains, books up to 6 passengers per PNR, pays online, cancels, views PNR status
Reservation Clerk	Human, railway staff	Books and cancels tickets at the counter on behalf of a passenger, accepts cash, prints tickets
Administrator	Human, railway staff	Adds trains, stations, routes, coaches, fares, and daily schedules; marks a run as cancelled or delayed; views reports
Payment Gateway	External system	Receives a payment request, returns success or failure with a transaction reference
Notification Service	External system	Sends SMS and email messages composed by the RRS

4. In-scope functions by module

No.	Module	Functions included in version 1.0	Main actor
1	User Management	Register with name, email, mobile, and password; login and logout; roles Passenger, Clerk, Administrator; change password; update profile	Passenger, Administrator
2	Train and Schedule Management	Add and edit trains (train number, name, type, coaches); add stations; define route with stop sequence, arrival, departure, and distance; add coaches by class (SL, 3A, 2A, 1A) with seat count; set fare per km per class; create a schedule for a run date; mark a schedule cancelled or delayed	Administrator
3	Search and Availability	Search trains by source station, destination station, and date; show seat availability per class for the schedule; show fare for the selected class and distance	Passenger, Clerk
4	Booking	Enter passenger details (name, age, gender, berth preference) for up to 6 passengers; allocate seats from available seats in the chosen class; place the booking on the waitlist when no seat is free; generate a unique 10-digit PNR; show and print the ticket; PNR status enquiry	Passenger, Clerk
5	Cancellation and Refund	Cancel a booking by PNR before departure; compute refund by time left before departure (100 percent minus flat clerkage above 48 hours, 50 percent between 48 and 12 hours, nil under 12 hours); release the seats; promote the first waitlisted booking in order	Passenger, Clerk

No.	Module	Functions included in version 1.0	Main actor
6	Payment	Initiate a payment for the booking amount; send the request to the Payment Gateway; record confirmed or failed status with the transaction reference; record cash payments taken by the clerk; issue a payment receipt	Passenger, Clerk
7	Notification	Send booking confirmation, cancellation confirmation, and schedule change alerts by SMS and email through the Notification Service; keep a log of messages sent	System
8	Reports	Occupancy per train per schedule; revenue by date range and train; cancellations and refunds by date range	Administrator

5. Out of scope

Item	Reason for exclusion
Unreserved (general class) ticketing	Different business process with no seat allocation
Tatkal and premium quota booking	Quota rules vary by train and season; deferred to a later version
Concessions (senior citizen, student, disabled)	Needs document verification that the online system cannot do
Meal ordering, porter booking, and station facilities	Not part of reservation
Route maps, live train tracking, and GPS location	Needs live data feeds the railway does not provide to this project
Voice input and multilingual user interface	Version 1.0 is English only
Lost and found, complaints, and grievance handling	Separate customer service system
Mobile app (Android or iOS)	Version 1.0 is a responsive web application usable in a mobile browser
Loyalty points and wallet	No requirement from the client
Season tickets and passes	No requirement from the client

6. Constraints

Type	Constraint
Technical	Web application; server side in Python 3; relational database (MySQL); runs in current versions of Chrome, Firefox, and Edge on desktop and mobile browsers
Technical	Passwords stored only as salted hashes; all traffic over HTTPS
Technical	Card and net banking details are never stored in the RRS; only the Payment Gateway transaction reference is kept
Regulatory	Refund rules follow the published railway refund rules in force on the date of cancellation
Regulatory	Passenger identity data is retained only for the period the railway prescribes and is not shared with third parties
Performance	Search results within 3 seconds for up to 500 concurrent users
Performance	Seat allocation is atomic: two bookings can never receive the same seat on the same schedule
Performance	Availability of 99.5 percent per month, excluding a published maintenance window
Schedule	Delivery within 10 calendar months of approval (see Session 2 estimate)
Budget	Development cost within the budget estimated in Session 2

7. Assumptions

- The railway supplies the master list of stations, trains, routes, and fare rates before development starts.
- The Payment Gateway and Notification Service expose documented APIs, and test accounts are available during development.
- A booking is for one train, one schedule, one class, and at most 6 passengers; multi-leg journeys are booked as separate PNRs.

- A schedule cannot be booked after its departure time from the source station.
- Reservation Clerks have a railway login and work on a counter PC connected to the same server as online passengers.
- One database serves all counters and online users; there is no offline mode.

8. Deliverables

Session	Deliverable
1	Statement of scope (this document)
2	Effort and cost estimate
3	Software Requirements Specification
4	DFDs, ERD, and data dictionary
5	Modular design
6	User interface design
13	Working RRS software with source code and test data
14	Change request, impact analysis, and revised software

9. Success criteria

- A passenger can register, search, book, pay, and receive a confirmation for a real schedule without staff help.
- No two bookings ever hold the same seat on the same schedule (verified by a concurrent booking test).
- The refund computed for any cancellation matches the rule for the time left before departure in every test case.
- Waitlisted bookings are promoted in order when a confirmed booking is cancelled.
- The Administrator can produce the occupancy and revenue reports for any date range without manual calculation.
- All functions in section 4 pass their acceptance tests, and nothing in section 5 is implemented.

10. Summary

The RRS version 1.0 covers eight modules serving three human actors and two external systems. It handles reserved seat booking, cancellation, online payment, notification, and administrative reporting for a single shared database. Unreserved ticketing, quotas, concessions, and customer service functions are excluded. The estimate in Session 2, the SRS in Session 3, and the design in Sessions 4 to 6 must stay inside these boundaries.

Viva Questions

× Do not copy. Read for understanding and the viva

- **Q:** What is the difference between scope and requirements? **A:** Scope fixes the boundary (which functions are in and out); requirements describe each function in detail with inputs, outputs, and rules.
- **Q:** Why does the scope statement list what is excluded? **A:** Exclusions prevent scope creep and let the team estimate, design, and test a fixed set of functions.
- **Q:** Who are the actors of the RRS? **A:** Passenger, Reservation Clerk, Administrator, and two external systems: Payment Gateway and Notification Service.
- **Q:** Is the Payment Gateway part of the RRS? **A:** No. It is an external system the RRS talks to; the RRS never stores card details.
- **Q:** What is a constraint? Give one from the RRS. **A:** A limit the team cannot change, for example search results within 3 seconds for 500 concurrent users.
- **Q:** What is an assumption? What happens if it fails? **A:** Something taken as true without proof, such as the railway supplying station data; if it fails, scope changes through change control.
- **Q:** Why is the module list fixed in Session 1? **A:** Sessions 2, 5, and 13 count, design, and implement by module; changing the list later breaks traceability.
- **Q:** Where does the scope statement appear in the project plan? **A:** At the start, as the overall goal, before objectives, deliverables, milestones, and budget.

Common Mistakes

× Do not copy. Read for understanding and the viva

- Writing a feature wish list (maps, voice input, chat) instead of a bounded set of functions with an explicit out-of-scope list.

- Mixing scope with design: describing screens, tables, or algorithms that belong in Sessions 3 to 6.
- Naming actors inconsistently (customer, user, client for the same person); pick Passenger and keep it.
- Stating constraints without a number ("the system should be fast") that cannot be tested.
- Forgetting external systems; the Payment Gateway and Notification Service shape the design more than any screen.
- Copying scope from a different system (airline, bus) and leaving in functions the railway does not need.

Session Summary

■ Write in lab record

Submit in the lab record:

- The problem statement copied from the manual.
- The full statement of scope, sections 1 to 10, with all four tables (actors, in-scope modules, out of scope, constraints).
- The assumptions list and success criteria as written.
- A note at the end: "Actors, modules, and entity names in this document are reused unchanged in Sessions 2 to 6, 13, and 14."

✎ Edit this page

< Previous
Introduction

Next >
Session 2