PROJECT PLAN: Hotel Booking & Management System

Milestone table:

Milestone ID	Milestone Name	Description	Planned Date	Deliverables
M1	Documentation Completion	All core planning docs completed	Week 2	SRS, Risk Analysis, Architecture Doc, UML
M2	Backend Implementation	Django setup, models, APIs, booking logic	Week 6	Working backend code, database schema, initial tests
M3	Frontend + Integration	UI built in Figma/Django templates and integrated with backend	Week 10	Integrated system with working UI (search, book, login, etc.)
M4	Testing & Debugging	Run unit/integration tests and fix bugs	Week 12	Test reports, bug fixes, QA checklist
M5	Deployment & User Training	Deploy system on Linux server + write user guide	Week 14	Deployed system, training material, user manual

Task Breakdown for M1 (Patrick):

Task ID	Task Name	Duration	Dependencies	Deliverables
M1.T1	Draft Risk Analysis	2 days	None	Initial risk categories and mitigation
M1.T2	Finalize Risk Analysis Document	2 days	M1.T1	HTML/CSS templates
M1.T3	Contribute to Architecture Summary	1 day	M1.T2	Section on how design supports risk plan
M1.T4	Review and Edit Full Planning Docs	2 days	M1.T3	Edited version of SRS, Risk Architecture

Resource Allocation

Human Resources:

Name	Assigned Tasks	Notes
Burak	M3.T3, M3.T4, M3.T5	Focused on frontend logic and bug fixing
Patrik Strzelczyk	M1	Handled early documentation
Nurkyz Bolotbekova	M3.T1	UI/UX in Figma
Nigar Alkhasova	M3.T2	Template development

Tool Resources:

Tool	Used For
Figma	UI mockup design
Django	Frontend-backend integration
GitHub	Version control
Visual Studio Code / PyCharm	Development IDE
Postman	API testing
Browser Dev Tools	UI debugging

Risk Management Integration:

Risk (ID & Description)	Affected Tasks / Areas
Risk 1: Payment gateway fails	M3.T3 (API integration), M3.T5 (UI Testing), M4 tasks (system-level testing)
Risk 2: Server downtime or hosting failure	M5 deployment tasks — especially final delivery and training
Risk 3: Inadequate user adoption	M3.T1–T5 (UI flow, design clarity), M5 (user training materials)
Risk 4: Budget overrun	M2 (backend overbuild), M3 (excessive front-end features), M4 (prolonged debugging cycles)
Risk 5: GDPR/non-compliance	M2 (data handling logic), M3.T4 (form validation), M4 (testing secure data handling)
Risk 6: Developer becomes unavailable	All milestones — especially task reassignment in M2/M3 if backend/front-end leads drop out

COMPLETE VALIDATION & TESTING SECTION

Test scope and exclusions:

Component/Feature	In Scope	Out of Scope	Associated Requirements
User Interface (UI)	Test all UI elements (forms, navigation, flow)	Visual design tweaks, color palettes	Room search, booking, login/logout, form validation
Booking Engine	Test room availability, booking, cancellation logic	Load testing with >1000 users	Make/cancel bookings, seasonal pricing, summary before confirm
Authentication	Login, logout, password reset	Security penetration testing	Secure login, password recovery, encryption
Admin Panel	Room updates, check-ins, reports	Deep analytics, financial dashboards	Admin update availability, view reports
Payment Integration	Test payment flows (e.g. Stripe, PayPal)	Fraud detection, PCI audits	Multi-payment method handling, booking completion

Test Cases

Test case 1

Test Scenario: User selects check-in and check-out dates using the calendar picker

Requirement Tested:
The system shall allow users to choose check-in and check-out dates from a visual calendar component.

Preconditions:

- The listing/search page is loaded.
- The date picker input fields ("Check in date", "Check out date") are visible.

Steps:

1. Click on the “Check in date” input field.
2. From the calendar popup, select a date in June 2025 (e.g., June 10).
3. Click on the “Check out date” input field.
4. Select a date in June 2025 that is later than the check-in (e.g., June 14).
5. Click “Done” to close the calendar.

Expected Outcome:

- The selected check-in and check-out dates appear correctly in their respective fields.
- The system does not allow selecting a check-out date that is before the check-in date.
- The calendar closes after the selection.

- The "Search" button remains enabled for further action.

Priority: Medium

Test Case 2:

Test Scenario: User enters payment details and proceeds to final confirmation

Requirement Tested:
The system shall allow users to input payment information and proceed to the final confirmation step.

Preconditions:

- The user has selected a room and reached Step 3 of the booking process.
- The payment page is visible and active in the Figma prototype.

Steps:

1. Select the “Credit/Debit Card” option (should be pre-selected by default).
2. Click into the “Bank Card No” field and enter mock card digits (e.g., 1234 5678 9012 3456).
3. Enter a name in the “Cardholder Name” field.
4. Enter expiration date and CVV in their respective fields.
5. (Optional) Click the promo code dropdown and choose an option.
6. Click the “Next Step: Final Confirmation” button.

Expected Outcome:

- The input fields accept user-typed text.
- Card and promo code fields are visually populated with user input.
- Upon clicking “Next Step: Final Confirmation,” the prototype navigates to the final screen or success message
- No error messages appear unless the fields are empty