NexaCare Solutions: Testing and Release Planning Kit

Prepared by: [Faisal Ahmed]
Role: Systems Analyst
Project: Appointment Scheduling Platform – Version 3.0 Release
Date: [7th October, 2025]

Project scenario

NexaCare Solutions is a healthcare technology company supporting over 300 clinics across North America. The organization is preparing to launch Version 3.0 of its web-based appointment scheduling platform.

This version introduces new features that aim to improve usability, patient engagement, and clinic operations. As the Systems Analyst, your responsibility is to coordinate and document a complete testing and release plan to ensure a safe, compliant, and successful rollout.

Key features in version 3.0

- SMS reminders
- Real-time doctor availability
- Calendar integration with Google and Outlook
- Redesigned user interface

Key considerations

- HIPAA compliance (over 1.2M patient records)
- First-time integration with third-party SMS provider
- Diverse client configurations (from basic to API-integrated systems)
- Pilot program for 20 clinics prior to full release
- Scheduled deployment window: Saturday, 12:00 AM – 6:00 AM BST

Test strategy outline

Overall approach

Types of testing and rationale

Type of Testing	Purpose / Rationale
Functional Testing	Evaluate SMS reminders, real-time doctor availability, calendar integrations, and the redesigned UI work as intended.
Regression Testing	Confirm that existing features (appointment booking, patient records, notifications) are not broken by the new release.
User Acceptance Testing (UAT):	Pilot clinics (20 selected) will validate that features meet real-world operational needs.
Performance Testing	Assess API response times, SMS delivery rates, and system scalability under peak usage.
Security & Compliance Testing	Ensure patient data is encrypted, access controls are enforced, and HIPAA compliance is maintained.

Feature test priorities

1. SMS reminders and calendar integration (new systems, higher risk)

2. Core scheduling functionality (crucial business impact)

3. UI refresh (usability and accessibility focus)

Testing objectives

Make sure that the NexaCare v3.0 platform operates seamlessly, integrates correctly with external systems, maintains data integrity, and provides a smooth user experience for clinics and patients.

Tools and frameworks

Testing Tools:

- *Selenium* – for automated functional and regression testing of the web interface

- *Postman* – for API testing and validation of SMS and calendar integration endpoints

CI/CD and Deployment:

- *Jenkins* – for automated build and deployment pipelines
- *GitHub Actions* – for version control integration and continuous delivery

Monitoring and Logging:

- *Grafana* – for system performance and uptime dashboards
- *Datadog* – for real-time alerting and error tracking

Test Management and Collaboration:

- *Jira* – for test case tracking, defect reporting, and release coordination

Traceability matrix

Requirement ID	Requirement Description	Test Case ID(s)	Coverage Status	Priority / Risk
RC_001	Verify 24h and 1h reminder sends correctly	TC_001	Covered	High
RC_002	Verify clinic can update availability instantly	TC_002	Covered	High
RC_003	Calendar integration with Google Calendar and Outlook	TC_003	Covered	Medium
RC_004	Book an appointment with valid patient info	TC_004	Covered	High
RC_005	Verify patient records migrated correctly	TC_005	Covered	High

Annotated test plans

Feature: [SMS Reminders]

Test objective:
[Verify that automated text reminders are sent to patients 24 hours and 1 hour before their scheduled appointments.]

Preconditions:

1. Valid patient phone numbers are stored in the system.

2. Third-party SMS service API credentials are active.

Test cases

Step	Test Action	Expected Result	Risks / Notes
1	Schedule a patient appointment	Appointment saved successfully	—
2	Wait for 24-hour trigger time	SMS reminder sent to patient	Delay if third-party API lags
3	Wait for 1-hour trigger time	SMS reminder sent again	Duplicate reminders if API retries
4	Cancel appointment	SMS notifications stop	Must validate cancellation logic

(Repeat this section for each major feature such as SMS reminders, calendar integration, real-time availability, etc.)

Feature: [Calendar Integration]

Test objective:
[Validate that appointments sync correctly with Google and Outlook calendars.]

Preconditions:

1. User has authorized Google/Outlook integration.

2. Appointment scheduling module active.

Test cases

Step	Test Action	Expected Result	Risks / Notes
1	Book a new appointment	Event created in	API token expiry

		external calendar	
2	Reschedule appointment	Event time updates correctly	Time zone mismatch
3	Reschedule appointment	Event time updates correctly	Time zone mismatch
4	Cancel appointment	Event removed from external calendar	Deletion delay

Feature: [Real-Time Doctor Availability]

Test objective:
[Ensure clinics can instantly update and share doctor availability across the platform.]

Preconditions:

1. Clinic staff logged in with valid credentials.

2. Network connection active.

Test cases

Step	Test Action	Expected Result	Risks / Notes
1	Update a doctor's availability slot	Changes visible immediately in portal	API latency
2	Simulate multiple users updating same doctor	Only latest update retained	Risk of overwrite
3	View updates from patient side	Updated availability reflected	Cache delay risk

Deployment checklist

Pre-deployment tasks

Task Description	Owner / Team	Est. Time	Dependencies / Notes
Validate final UAT approval from	QA Team	2 hrs	

pilot clinics			UAT completion required
Backup production database	DevOps	1 hr	Verify snapshot integrity
Notify clinics about downtime window	Support	30 mins	Send email notifications
Verify SMS and calendar API credentials	DevOps	1 hr	Third-party access check

Deployment window tasks

Task Description	Owner / Team	Time Window	Prerequisites
Deploy release v3.0 to production servers	DevOps	12:30–2:00 AM BST	Backup confirmed
Run smoke tests (login, booking, SMS)	QA	2:00–3:00 AM BST	Deployment successful
Validate API integrations	QA and DevOps	3:00–4:00 AM BST	Smoke tests passed
Conduct pilot verification	Support	4:00–5:30 AM BST	Pilot clinics available

Post-deployment tasks

Task Description	Owner / Team	Est. Time	Dependencies / Notes
Pilot clinics available	DevOps	2 hrs	Grafana, Datadog dashboards
Collect pilot feedback	Support	24 hrs	Post-launch
Prepare release summary report	Product Team	1 day	Input from all teams

Go / No-Go criteria

- All smoke tests pass successfully.

- No critical or high-severity issues in error logs.
- Database migration validation complete.
- Uptime ≥ 99% during pilot phase.

Rollback strategy

Conditions that trigger rollback:

- SMS or calendar integrations failing platform-wide.
- Significant data loss or corruption.
- Core scheduling module unavailable for more than 30 minutes.

Rollback procedure

Step	Description	Responsible Team / Person
1	Halt incoming traffic to v3.0	DevOps
2	Restore previous version database snapshot	DevOps
3	Redeploy version 2.x	DevOps
4	Verify system accessibility	QA
5	Notify stakeholders and clinics	Product Manager

Communication Plan:

- Internal notification via Slack and Jira.
- External notice to clinics via email and status page update.

Estimated Rollback Time:
Best case: 1 hour | Worst case: 3 hours

Data migration plan

Systems Involved:

- **Source System:** Legacy NexaCare Appointment Database (v2.x)

- **Destination System:** NexaCare v3.0 Cloud Platform

Field mapping

Source Field	Destination Field	Transformation / Logic
first_name + last_name	full_name	Concatenate with space
phone	contact_number	Validate number for SMS format
appointment_date	appointment_time_marker	Convert to UTC format
doctor_availability	availability_status	Map Yes/No to True/False

Migration volume

Approx. 1.2 million records, migrated in 5 batches of 240,000 each.

Validation steps

- Pre-migration sanity checks
- Migration logs review
- Post-migration field and count verification
- Exception reporting and remediation plan

Post-deployment monitoring summary

Key metrics to monitor

Metric	Threshold / SLA	Monitoring Tool / Method
API Response Time	< 300 ms	Grafana
Error Rate	< 2% per hour	Datadog
SMS Delivery Success	≥ 99%	SMS Provider Dashboard
Database CPU Usage	< 80%	AWS CloudWatch

Uptime	≥ 99.9%	Uptime Robot
User Login Failures	< 1% per hour	Application Logs

Monitoring responsibilities

Team / Person	**Monitoring Role**	**Frequency / Schedule**
DevOps	System health, uptime, resources	Continuous (24/7)
QA	Functional smoke tests	Every 4 hrs post-deployment (first 24 hrs)
Support	User issues and feedback tracking	Daily
Product	Review dashboards, summarize KPIs	Weekly

Response plan

- **Step 1:** Alert triggered → Auto-notify DevOps via PagerDuty.
- **Step 2:** DevOps investigates root cause (API, DB, infra).
- **Step 3:** Escalate to relevant teams if threshold breached.
- **Step 4:** Document incident in Jira and communicate via Slack.
- **Step 5:** Update stakeholders with resolution summary.