

WhatsApp Package Notification System

Modular Design & Waterfall Development Plan

System Architecture Overview



Module Development Sequence (Waterfall Approach)

Phase 1: Foundation Modules (Weeks 1-8)

Module 1: Database Design & Implementation

Development Order: FIRST Duration: 2 weeks

Reason for Priority:

- Foundation for all other modules
- Data structure must be stable before building dependent modules
- Changes later would cascade through entire system
- Enables parallel development of other modules once complete

Components:

Database Schema:

1. customers

- customer_id (PK)
- first_name
- last_name
- phone_number (with country code)
- email
- preferred_language
- whatsapp_opt_in (boolean)
- created_at
- updated_at

2. packages

- package_id (PK)
- tracking_number (unique)
- customer_id (FK)
- description
- weight
- declared_value
- origin_country
- arrival_date
- status (arrived, notified, delivery_confirmed, picked_up, delivered)
- delivery_preference (null, home_delivery, pickup)
- created_at
- updated_at

3. notifications

- notification_id (PK)
- package_id (FK)
- customer_id (FK)
- message_content
- whatsapp_message_id
- sent_at
- delivered_at
- read_at
- status (queued, sent, delivered, read, failed)
- failure_reason

4. customer_responses

- response_id (PK)
- notification_id (FK)
- customer_id (FK)
- package_id (FK)
- response_text
- parsed_intent (delivery, pickup, query, other)

- received_at
- processed (boolean)

5. chat_conversations

- conversation_id (PK)
- customer_id (FK)
- message_type (incoming, outgoing)
- message_content
- sender (customer, chatbot, agent)
- intent_detected
- confidence_score
- escalated_to_human (boolean)
- timestamp

6. users (staff)

- user_id (PK)
- username
- password_hash
- full_name
- email
- role (admin, operator, viewer)
- last_login
- active (boolean)
- created_at

7. message_templates

- template_id (PK)
- template_name
- language_code
- content
- variables (JSON array)
- whatsapp_approved (boolean)
- category (notification, follow_up, confirmation)
- created_at
- updated_at

8. chatbot_knowledge_base

- kb_id (PK)
- question_pattern
- answer_content
- category (location, hours, pricing, procedures)
- language_code
- priority
- active (boolean)
- usage_count
- last_updated

9. system_logs

- log_id (PK)
- user_id (FK, nullable)
- action_type
- entity_type
- entity_id
- details (JSON)
- ip_address
- timestamp

Deliverables:

- Complete database schema
 - Database creation scripts
 - Index optimization plan
 - Backup and recovery procedures
 - Data migration scripts (if integrating existing data)
-

Module 2: User Authentication & Authorization

Development Order: SECOND Duration: 2 weeks

Reason for Priority:

- Required before any staff can access the system
- Security foundation must be solid from the start
- Independent of business logic, can be developed while Module 3 progresses
- Protects all subsequent modules

Components:

2.1 Authentication System

- Login/logout functionality
- Password hashing (bcrypt or Argon2)
- Session management
- Password reset mechanism
- Account lockout after failed attempts
- Two-factor authentication (optional Phase 2)

2.2 Authorization System

- Role-based access control (RBAC)
- Permission matrix:

Admin: Full system access

- User management
- Template management
- System configuration
- All operator permissions

Operator: Day-to-day operations

- Send notifications
- View customer data
- Update package status
- View reports

Viewer: Read-only access

- View notifications
- View reports
- No modifications allowed

2.3 Audit Logging

- Log all authentication attempts
- Log all user actions
- Track data access and modifications

Deliverables:

- Login page (web interface)
- User management interface
- Role assignment system
- Audit log viewer
- Security documentation

Module 3: Customer & Package Management

Development Order: THIRD Duration: 3 weeks

Reason for Priority:

- Core business data management
- Required before notifications can be sent
- Provides data entry interface for staff
- Foundation for notification targeting

Components:

3.1 Customer Management Interface

- Add/edit/view customer records
- Bulk customer import (CSV/Excel)
- Phone number validation and formatting
- Search and filter customers
- Customer history view
- WhatsApp opt-in management

3.2 Package Management Interface

- Register new package arrivals
- Scan tracking numbers (barcode/QR)
- Manual tracking number entry
- Link packages to customers
- Update package status
- Search packages by tracking number/customer
- Package history and status tracking

3.3 Data Validation

- Phone number format validation (E.164)
- Duplicate detection
- Required field enforcement
- Data integrity checks

3.4 Integration Preparation

- API endpoints for external system integration
- Webhook receivers for existing package tracking system
- Data synchronization logic

Deliverables:

- Customer management dashboard
 - Package registration interface
 - Search and filter functionality
 - Bulk import tools
 - API documentation for integrations
-

Module 4: WhatsApp Business API Integration

Development Order: FOURTH Duration: 1 week

Reason for Priority:

- Technical integration must be established before notification features
- API testing and credential configuration needed early
- Identifies rate limits and constraints
- Relatively independent, can be developed in parallel with Module 3

Components:**4.1 API Connection Setup**

- WhatsApp Business API credentials configuration
- Webhook setup for receiving messages
- API rate limit handling
- Connection health monitoring
- Error handling and retry logic

4.2 Message Sending Infrastructure

- Send text messages via API
- Send template messages
- Send interactive messages (buttons)
- Message queue implementation
- Batch sending with rate limiting

4.3 Message Receiving Infrastructure

- Webhook endpoint for incoming messages
- Message parsing and routing
- Delivery status updates (sent, delivered, read)
- Read receipt processing

4.4 Testing Framework

- Test message sending
- Test message receiving
- Simulate various WhatsApp responses
- Load testing for bulk operations

Deliverables:

- WhatsApp API integration library
 - Configuration management interface
 - Message queue system
 - Webhook handlers
 - Integration testing suite
-

Phase 2: Core Business Logic (Weeks 9-16)

Module 5: Notification Engine

Development Order: FIFTH Duration: 3 weeks

Reason for Priority:

- Core business value - the main purpose of the system
- Depends on Modules 1, 3, and 4 being complete
- Most critical functionality for solving the original problem

Components:

5.1 Notification Trigger System

- Automatic notification on package arrival
- Manual notification triggering
- Scheduled notification sending
- Notification preview before sending

5.2 Bulk Notification Handler

- Select multiple customers/packages
- Queue management for bulk operations
- Progress tracking
- Estimated completion time calculation
- Success/failure reporting

5.3 Targeted Messaging

- Filter by arrival date
- Filter by package status
- Filter by delivery preference
- Filter by customer location
- Custom recipient selection

5.4 Message Composition

- Template selection
- Variable substitution (name, tracking #, etc.)
- Message preview
- Multi-language support
- Custom message creation for specific scenarios

5.5 Notification Status Tracking

- Real-time status updates
- Delivery confirmation
- Read receipts
- Failed message handling
- Retry logic for failures

Deliverables:

- Notification dashboard
- Bulk sending interface
- Targeted messaging filters

- Message composition tool
 - Status tracking display
 - Notification history logs
-

Module 6: Response Handler & Processing

Development Order: SIXTH Duration: 2 weeks

Reason for Priority:

- Completes the notification workflow loop
- Depends on Module 5 (notifications must be sent first)
- Critical for capturing customer delivery preferences

Components:

6.1 Response Collection

- Receive incoming WhatsApp messages
- Link responses to original notifications
- Timestamp all interactions
- Handle multiple responses from same customer

6.2 Intent Recognition

- Keyword detection (deliver, pickup, collect, etc.)
- Natural language parsing (basic)
- Confidence scoring
- Handle ambiguous responses

6.3 Status Updates

- Automatically update package status based on response
- Flag unclear responses for manual review
- Update customer delivery preference

6.4 Follow-up Logic

- Identify non-responsive customers
- Schedule automatic reminders

- Escalation rules (after X days)

6.5 Manual Override

- Staff can manually interpret responses
- Override automated status updates
- Add notes to customer interactions

Deliverables:

- Response processing engine
 - Intent recognition system
 - Status update automation
 - Follow-up scheduling system
 - Manual review interface
-

Module 7: Message Template Management

Development Order: SEVENTH Duration: 2 weeks

Reason for Priority:

- Enhances Module 5 but not critical for MVP
- WhatsApp requires template approval (can take days)
- Templates can be hardcoded initially, this module adds flexibility

Components:

7.1 Template Creation Interface

- Create new templates
- Define variable placeholders
- Preview with sample data
- Support multiple languages

7.2 WhatsApp Template Submission

- Format templates per WhatsApp requirements
- Submit for approval
- Track approval status

- Handle rejections and resubmissions

7.3 Template Library

- Browse all templates
- Search and filter
- Categorize by purpose
- View usage statistics

7.4 Version Control

- Track template changes
- Roll back to previous versions
- Compare versions

Deliverables:

- Template creation interface
 - Template approval workflow
 - Template library browser
 - Version history tracking
-

Phase 3: Advanced Features (Weeks 17-24)

Module 8: Chatbot Engine

Development Order: EIGHTH Duration: 4 weeks

Reason for Priority:

- Enhancement feature, not core MVP requirement
- Most complex module requiring NLP/AI
- Depends on stable notification system being operational
- Can be phased: rule-based first, then AI enhancement

Components:

8.1 Conversation Manager

- Maintain conversation context
- Session management

- Conversation history tracking
- Multi-turn dialogue support

8.2 Intent Recognition Engine

- Rule-based pattern matching (Phase 1)
- Machine learning classification (Phase 2)
- Confidence scoring
- Fallback handling

8.3 Response Generation

- Template-based responses
- Dynamic content insertion
- Personalization
- Multi-language support

8.4 Knowledge Base Management

- FAQ database
- Question-answer pairs
- Categorization
- Easy updating without coding

8.5 Integration with Notification System

- Access package tracking data
- Provide real-time status updates
- Link chatbot conversations to customer records

8.6 Escalation System

- Detect when human intervention needed
- Alert staff of escalated conversations
- Seamless handoff to human agent
- Conversation history transfer

8.7 Quick Reply Buttons

- Common question shortcuts

- Delivery preference buttons
- Yes/No confirmations
- Menu navigation

8.8 Chatbot Analytics

- Track common queries
- Measure resolution rate
- Identify knowledge gaps
- Customer satisfaction scoring

Submodules (Priority Order within Module 8):

8A. Rule-Based FAQ System (Week 1-2)

- Pattern matching for exact questions
- Keyword detection
- Static responses
- Quick wins for common queries

8B. Conversation Context Manager (Week 2)

- Track conversation state
- Remember previous messages
- Context-aware responses

8C. Escalation Logic (Week 3)

- Identify complex queries
- Sentiment analysis (basic)
- Human handoff mechanism

8D. Knowledge Base Interface (Week 3-4)

- Admin panel for FAQ management
- Test chatbot responses
- Analytics dashboard

8E. AI/ML Enhancement (Future Phase)

- Training data collection

- Machine learning model
- Continuous improvement

Deliverables:

- Functional chatbot system
 - Knowledge base with 50+ FAQ entries
 - Admin interface for KB management
 - Escalation workflow
 - Chatbot analytics dashboard
 - Training documentation
-

Module 9: Reporting & Analytics

Development Order: NINTH Duration: 2 weeks

Reason for Priority:

- Business intelligence feature, not critical for operations
- Requires historical data to be meaningful
- Depends on all operational modules generating data

Components:**9.1 Dashboard Widgets**

- Total notifications sent (daily/weekly/monthly)
- Delivery success rate
- Response rate
- Average response time
- Pending packages count
- Chatbot interaction statistics

9.2 Report Generation

- Notification performance report
- Customer engagement report
- Package status summary
- Chatbot effectiveness report

- Comparison with email/SMS (if data available)

9.3 Data Visualization

- Charts and graphs
- Trend analysis
- Performance metrics
- Export to PDF/Excel

9.4 Custom Reports

- Date range selection
- Filter by customer/package/status
- Scheduled report generation
- Email delivery of reports

Deliverables:

- Analytics dashboard
 - Pre-built report templates
 - Custom report builder
 - Data export functionality
 - Scheduled reporting system
-

Phase 4: Optimization & Enhancement (Weeks 25-28)

Module 10: System Administration & Monitoring

Development Order: TENTH (LAST) Duration: 2 weeks

Reason for Priority:

- System must be operational before monitoring is critical
- Optimization requires usage data
- Final polish after all features are implemented

Components:

10.1 System Health Monitoring

- API uptime monitoring

- Database performance metrics
- Queue status monitoring
- Error rate tracking
- Alert system for critical issues

10.2 Configuration Management

- System settings interface
- WhatsApp API configuration
- Rate limit settings
- Business hours configuration
- Notification schedules

10.3 Backup & Recovery

- Automated backup scheduling
- Backup verification
- Restore procedures
- Disaster recovery plan

10.4 Performance Optimization

- Database query optimization
- Caching implementation
- Load balancing (if needed)
- API call optimization

10.5 System Logs & Debugging

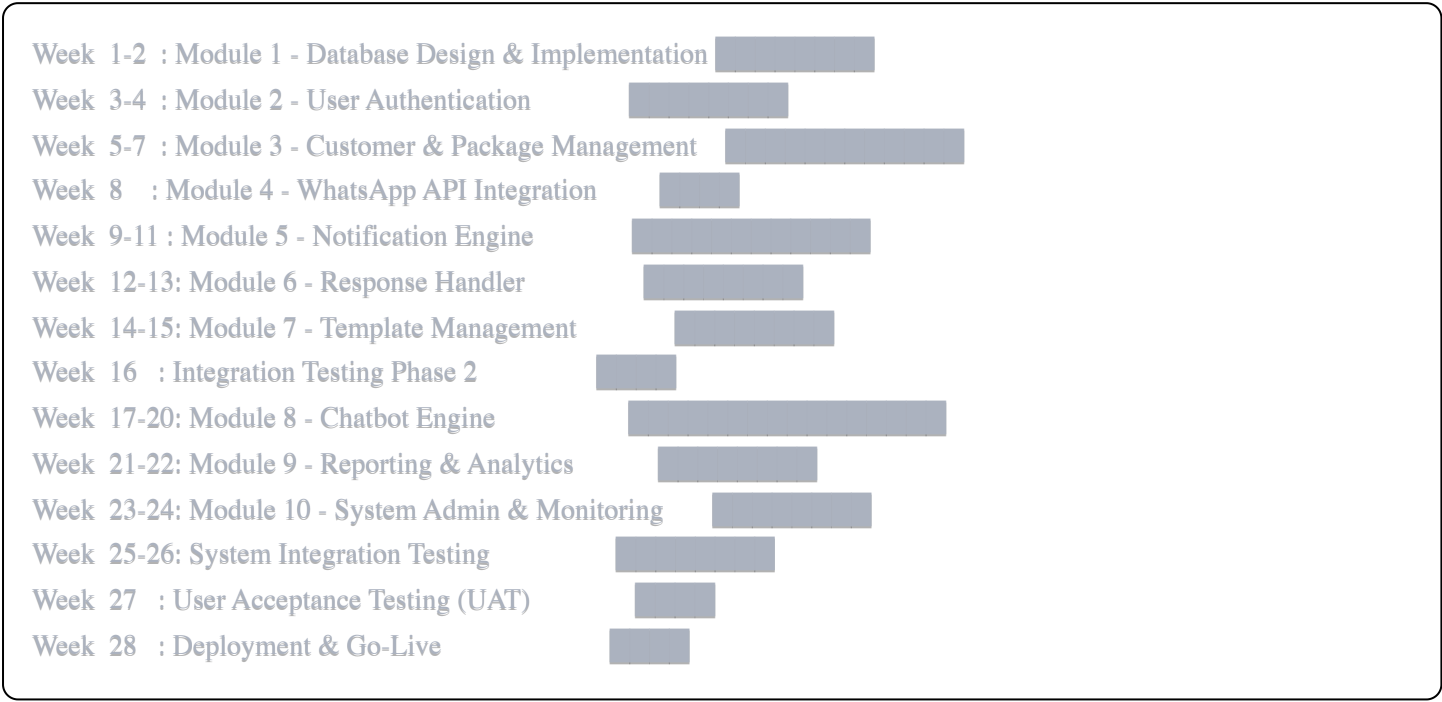
- Centralized log management
- Log search and filtering
- Error tracking and debugging
- Performance profiling

Deliverables:

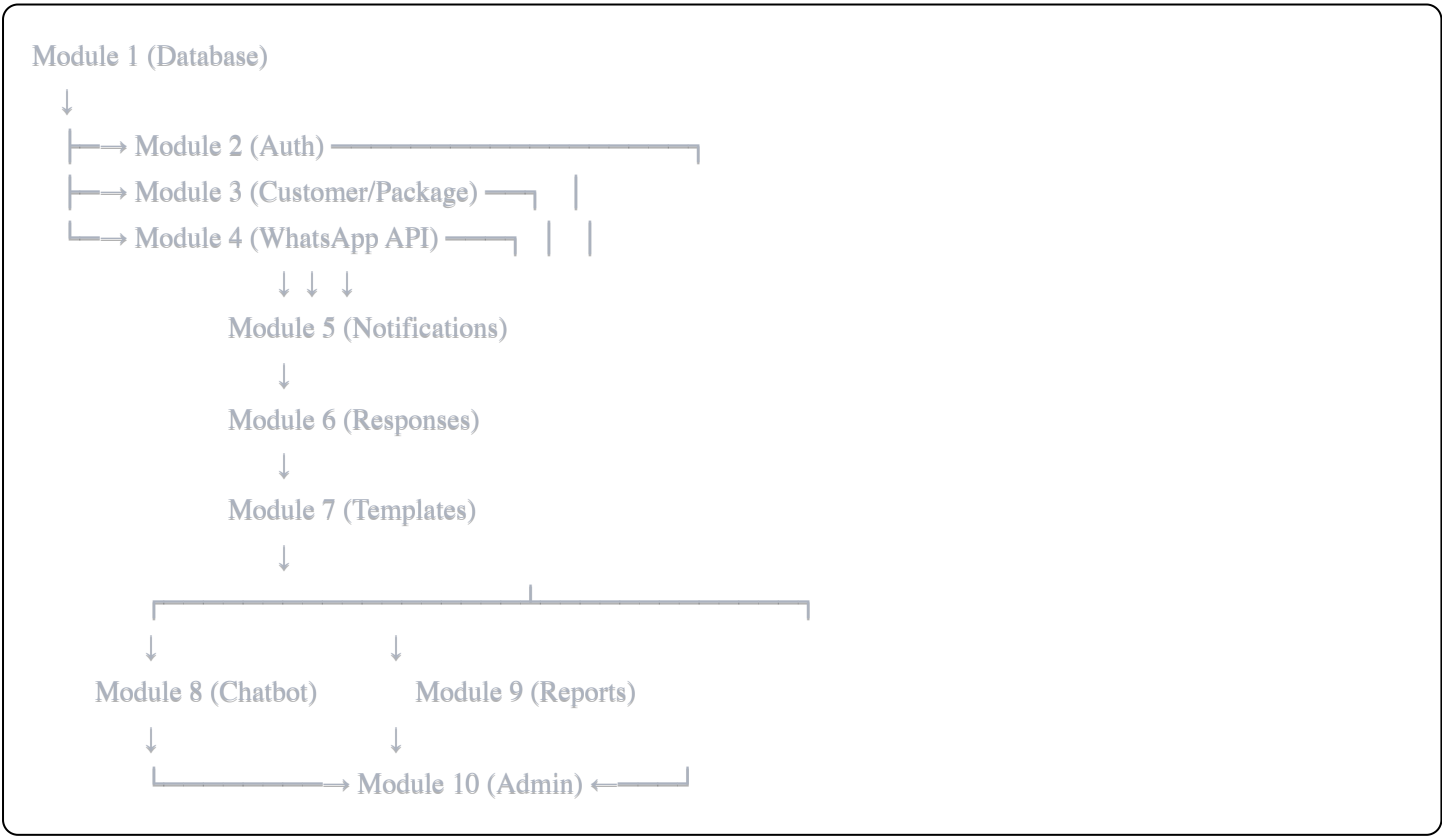
- System admin panel
- Monitoring dashboard

- Backup/restore tools
- Performance optimization report
- System documentation

Waterfall Development Timeline Summary



Module Dependencies Diagram



Critical Success Factors for Waterfall Approach

1. Complete Requirements Before Development

- All requirements must be documented before coding
- Changes mid-development are costly
- Get stakeholder sign-off on each module spec

2. Rigorous Testing at Each Phase

- Unit testing after each module
- Integration testing between dependent modules
- Regression testing after changes

3. Documentation Throughout

- Technical documentation for developers
- User manuals for staff
- API documentation for integrations

4. Milestones & Quality Gates





- Each module must pass quality checks before next module begins
- Code reviews at module completion
- Security audits before production



5. Risk Management

- WhatsApp API approval may take time - start early
- Have fallback plans for failed integrations
- Budget buffer for unforeseen issues

Minimum Viable Product (MVP) Scope

For fastest time-to-value, MVP should include:

-  Module 1: Database
-  Module 2: Authentication
-  Module 3: Customer/Package Management
-  Module 4: WhatsApp Integration

-  Module 5: Notification Engine
-  Module 6: Response Handler

MVP Timeline: 13 weeks

Defer to Phase 2:

- Module 7: Template Management (use hardcoded templates initially)
 - Module 8: Chatbot (manual customer service initially)
 - Module 9: Advanced Reporting (basic reports sufficient)
 - Module 10: Advanced Admin (basic admin tools sufficient)
-

Testing Strategy

Module Testing (During Development)

- Unit tests for each function
- Integration tests between components
- Performance tests for bulk operations

Phase Testing (End of Each Phase)

- Integration testing across modules
- End-to-end testing of workflows
- Security testing
- User acceptance testing

System Testing (Before Deployment)

- Full system integration test
 - Load testing with production-like data
 - Disaster recovery testing
 - Security penetration testing
-

Deployment Strategy

1. Pilot Phase (Week 29-30)

- Deploy to limited user group (10-20 customers)
- Monitor closely for issues

- Gather user feedback

2. Staged Rollout (Week 31-32)

- Gradually increase customer base
- 25% → 50% → 75% → 100%
- Monitor system performance at each stage

3. Full Production (Week 33)

- All customers migrated
 - 24/7 monitoring active
 - Support team trained and ready
-

Post-Deployment Support Plan

Week 1-4 (Hypercare)

- Daily monitoring
- Immediate bug fixes
- Daily stakeholder updates

Week 5-12 (Stabilization)

- Weekly monitoring reviews
- Planned updates and improvements
- User training sessions

Week 13+ (Business as Usual)

- Monthly review meetings
- Quarterly feature enhancements
- Continuous optimization