FPT UNIVERSITY

GenCare: Smart Gender Healthcare System

Software Requirement Specification

Class Code: SWR302

Group Code: G3

Ho Chi Minh City, June 30, 2025

RECORD OF CHANGE

*A - Added M - Modified D - Deleted

Effective Date	Changed Items	A/M/D	Change Description	New Version
01/06/2025	SRS baseline	A	Submitted initial skeleton, automatic table of contents and placeholder sections	v0.1
05/06/2025	Process diagrams	A	Added detailed swim-lane diagrams for Appointment, Blog and STI workflows	v0.2
10/06/2025	Authentication chapter	M	Refined login requirements; introduced `email_verified` rule and updated sequence diagram	v0.3
15/06/2025	Sections 2 & 3	M	Completed Overall Description and Technology Stack, aligned with current codebase	v0.4
20/06/2025	Testing strategy	A	Documented Selenium end-to-end suite and Jest unit/integration plan	v0.5
25/06/2025	Performance specification	M	Re-calibrated response-time targets; limited avatar upload to 5 MB	v0.6
27/06/2025	NFR consolidation	M	Merged duplicate NFR chapters; standardised UI stack to Tailwind + shadcn/ui	v0.7
28/06/2025	Road-map features	A	Flagged Manager role, Pill Tracking and Redis rate-limiting as future scope (TBD)	v0.8
30/06/2025	Final QA sign-off	M	Proof-read entire document, fixed formatting, issued release candidate.	v1.0

SIGNATURE PAGE

ORIGINATOR: Tran Phu Tho Date _____

Position: Team Leader

REVIEWERS: Tran Ngoc Nhu Quynh Date 30/06/2025

Position: Project Reviewer

Pham Hoang Tuyet Ngan Date 14/07/2025

Position: Project Reviewer

Contents

1 INTRODUCTION

The introduction of the Software Requirements Specification (SRS) provides an overview of the entire SRS. It includes the purpose, scope, definitions, acronyms, abbreviations, references, and overview of the SRS.

The SRS document captures the complete software requirements for the Healthcare Consultation & Testing Platform system. This document captures all requirements using traditional, natural-language style requirements with use-case modeling. It consists of functional and non-functional requirements along with supporting technical specifications.

1.1 Purpose

This Software Requirements Specification (SRS) document specifies the complete requirements for the GenCare - Healthcare Consultation. The SRS fully describes the external behavior of the application and all identified subsystems. It also describes nonfunctional requirements, design constraints, and other factors necessary to provide a complete and comprehensive description of the requirements for the software.

This document is intended for:

- Development team members responsible for system implementation
- Quality assurance team for testing and validation
- Project stakeholders including healthcare professionals and administrators
- System integrators and maintenance personnel
- End users including patients, consultants, and healthcare staff

The GenCare platform is designed to provide:

- Online reproductive health consultation services between patients and certified healthcare consultants
- STI (Sexually Transmitted Infections) testing management and scheduling system
- Personal health record management with menstrual cycle tracking
- Educational blog content delivery for reproductive health awareness
- Multi-role user management system with healthcare-grade security compliance
- Real-time appointment scheduling and reminder system
- Comprehensive reporting and analytics dashboard

1.2 Definitions, Acronyms

This subsection provides the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS.

API Application Programming Interface

JWT JSON Web Token

STI Sexually Transmitted Infection

SRS Software Requirements Specification

UI User Interface

UX User Experience

RBAC Role-Based Access Control

HIPAA Health Insurance Portability and Accountability Act

OTP One-Time Password

REST Representational State Transfer

CRUD Create, Read, Update, Delete

E2E End-to-End Testing

QA Quality Assurance

CI/CD Continuous Integration/Continuous Deployment

Selenium Web browser automation tool for testing

WebDriver Browser automation API used by Selenium

MongoDB NoSQL document database

React JavaScript library for building user interfaces

Node.js JavaScript runtime environment

Express.js Web application framework for Node.js

Joi JavaScript validation library used for request schema enforcement

Redis In-memory key–value store used for caching, OTP and rate-limiting

Mongoose Object-Document Mapper (ODM) for MongoDB in Node.js

Tailwind CSS Utility-first CSS framework for styling the front-end

shadcn/ui Headless React component library built on Tailwind CSS

OpenAPI Standard, language-agnostic interface description for REST APIs

1.3 References

This subsection provides a complete list of all documents referenced elsewhere in the SRS. Each document is identified by title, report number (if applicable), date, and publishing organization.

1. Ministry of Health of Vietnam. *Circular 49/2018/TT-BYT on Guidelines for Laboratory Testing Activities in Medical Examination and Treatment.* Hanoi: Ministry of Health, 28 December 2018.
2. National Assembly of Vietnam. *Law on Network Information Security No. 86/2015/QH13.* Hanoi: National Assembly, 19 November 2015.
3. National Assembly of Vietnam. *Law No. 15/2023/QH15 on Medical Examination and Treatment.* Hanoi: National Assembly, 09 January 2023.
4. Government of Vietnam. *Decree 13/2023/ND-CP on Personal Data Protection.* Hanoi: Government Office, 17 April 2023.
5. Fielding, Roy Thomas. *Architectural Styles and the Design of Network-based Software Architectures.* Doctoral dissertation, University of California, Irvine, 2000.
6. Tailwind Labs Inc. *Tailwind CSS: Utility-First CSS Framework.* 2024. Available at: https://tailwindcss.com/
7. shadcn. *shadcn/ui: Headless React Component Library.* 2024. Available at: https://ui.shadcn.con
8. MongoDB Inc. *MongoDB Manual Release 7.0.* MongoDB Inc., 2024. Available at: https://docs.mongodb.com/
9. Meta Platforms Inc. *React Documentation (v18).* Meta Platforms Inc., 2024. Available at: https://react.dev/
10. OpenJS Foundation. *Node.js and Express.js Documentation (v18 / v4).* OpenJS Foundation, 2024. Available at: https://nodejs.org/ and https://expressjs.com/
11. SeleniumHQ. *Selenium WebDriver Documentation.* SeleniumHQ, 2024. Available at: https://selenium-python.readthedocs.io/
12. Python Software Foundation. *Python Testing Best Practices.* Python Software Foundation, 2024. Available at: https://docs.python.org/3/library/unittest.html
13. Meta Platforms Inc. *Jest Testing Framework Documentation (v29).* Meta Platforms Inc., 2024. Available at: https://jestjs.io/
14. Centers for Disease Control and Prevention. *Sexually Transmitted Infections Treatment Guidelines, 2021.* MMWR Recommendations and Reports, Vol. 70, No. 4, 23 July 2021.
15. Habiba, Marwan, and Giuseppe Benagiano. "The duration of menstrual blood loss: historical to current understanding." Reproductive Medicine 4.3 (2023): 145-165.
16. Menstrual cycle: What's normal, what's not. Available at: Mayo Clinic
17. Reference from: https://www.pharmacity.vn/kham-pha-ve-fertile-window-va-cach-xac-dinh-chinh-xac-thoi-gian-thu-thai.htm

18. Reference from: https://www.hopkinsmedicine.org/health/wellness-and-prevention/calculating-your-monthly-fertility-window

19. Reference from: https://www.plannedparenthood.org/learn/birth-control/birth-control-pill/how-do-i-use-the-birth-control-pill

2 Overall Description

2.1 Product Perspective

GenCare is an *all-in-one* tele-healthcare platform built on a client-server architecture:

- **Frontend:** Single-Page Application using React and TypeScript.
- **Backend:** Node.js (Express) with TypeScript, MongoDB for persistent storage, Redis for caching, OTP storage, and rate-limiting.
- **Security:** JWT (access & refresh tokens) and Google OAuth.
- **Third-party integrations:** Gmail SMTP (OTP, notifications), Google Meet (online consultation rooms), and Momo (payment gateway).
- **Code organisation:** Controller → Service → Repository → Model, enabling future micro-service extraction.
- **Quality assurance:** Selenium test suites with CI ready scripts.
- **AI-Powered Health Assistant:** Intelligent chatbot for 24/7 health consultation and guidance using Azure OpenAI Chat Model with specialized healthcare knowledge base.

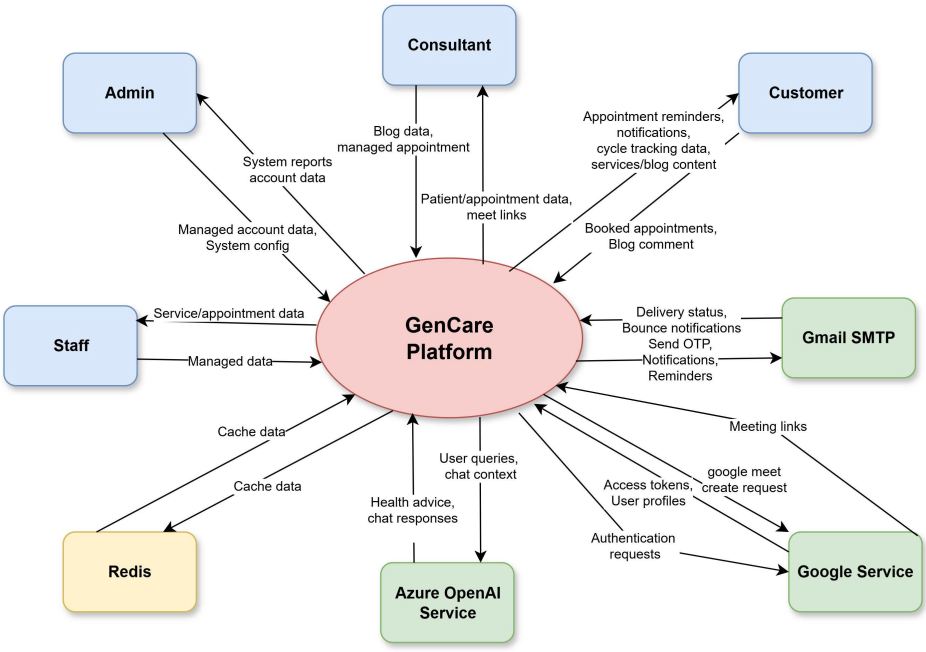

Figure 1: Context Diagram of Gencare system.

2.2 Business Process

The platform supports three primary business flows (swim-lanes), where all flows trigger notification events through a central `emailNotificationService` and are logged in Redis/MongoDB for reporting:

2.2.1 Booking Appointment

Booking Appointment The appointment booking process is illustrated in Figure 2, which demonstrates the interaction between customers and consultants throughout the booking workflow. As shown in the swimlane diagram, the process begins when a customer selects a preferred consultant and time slot, followed by system validation and consultant confirmation.

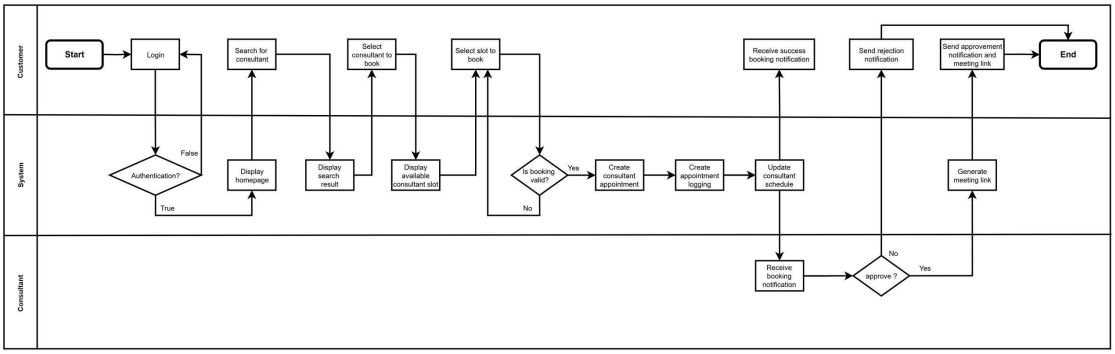

Figure 2: Booking Appointment Swimlane Diagram

Consultant Flow

1. Sign-in and manage personal schedule.
2. Review appointments and access auto-generated Google Meet links.
3. Publish professional blog posts.
4. Monitor feedback ratings.

Staff & Admin Flow

- **Staff:** clone or adjust consultant schedules, supervise appointments, access statistics.
- **Admin:** full CRUD over users, blogs, audit-log monitoring and system configuration.

2.2.2 Booking STI Test

Booking STI Test The STI test booking workflow is detailed in Figure 3, which outlines the complete process from booking date for testing, meeting offline consultant for selecting package/tests to create/update result, then ends with consultant confirmation. After that, the result will send to customer through email. The diagram shows how customers interact with the system to select schedule for their testing and how staffs and consultants interacts with their patients.

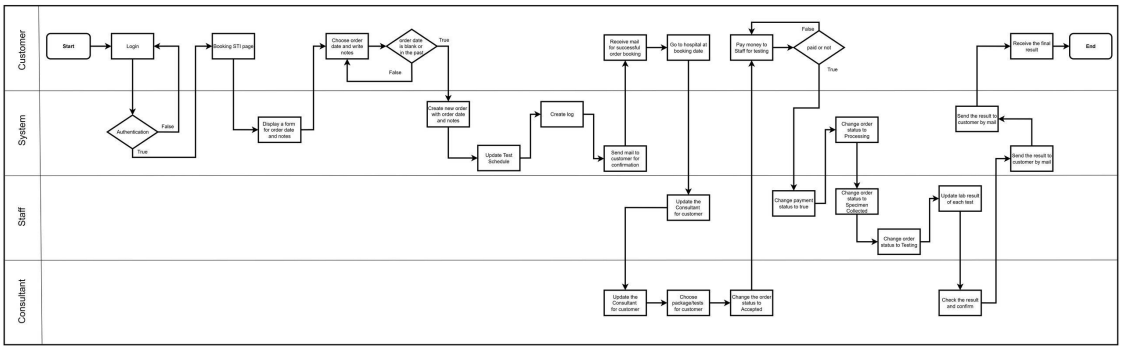

Figure 3: STI Test Booking Swimlane Diagram

Consultant Flow

1. Screening and test suggestion.
2. Check final results for confirmation.

Staff & Admin Flow

- **Staff flow:**
  1. Update consultant for customer.
  2. Manage the payment.
  3. Manage order status.
  4. Laboratory sample testing and write result.
- **Admin flow:**
  1. Full CRUD of STI packages and tests.
  2. Audit-log monitoring and system configuration.
  3. Tracking revenue sources

2.2.3 Menstrual Cycle and Pill Tracking Management Process

Menstrual Cycle management and Pill tracking The menstrual cycle and pill tracking management process is presented in Figures 4, 5. These diagrams illustrate how the system supports women's health monitoring through automated cycle tracking, prediction algorithms, and medication reminder capabilities.

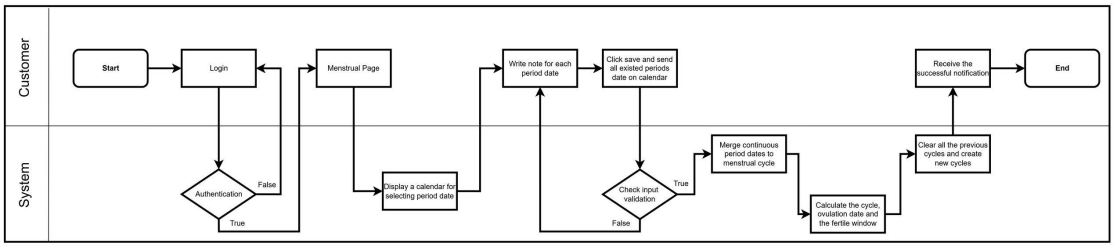

Figure 4: Menstrual Cycle Management Swimlane Diagram

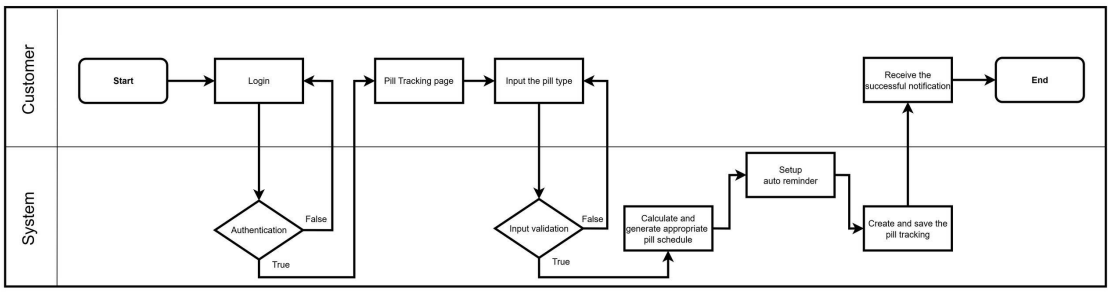

Figure 5: Pill Tracking Management Swimlane Diagram

Health Tracking Process

1. Customer inputs menstrual cycle data and pill-taking schedule.
2. System analyzes patterns and generates predictions for future cycles.
3. Automated reminders are sent for pill-taking and cycle tracking.
4. Healthcare insights and recommendations are provided based on data analysis.

Staff & Admin Flow

- **Staff:** Monitor health tracking compliance and provide support to customers.
- **Admin:** Configure tracking parameters, manage reminder settings, and access aggregate health statistics.

2.3 User Classes and Characteristics

Table 1: User classes and their main privileges

User Class	Characteristics / Goals	Primary Privileges
Guest	Unauthenticated, anonymous access	View landing page and public blogs
Customer	Patients; mobile-first usage	Sign-up, email verification, booking, cycle tracking, STI packages, feedback
Consultant	Doctors / specialists; busy schedules	Manage timetable, conduct consultations, create blogs, view personal feedback
Staff	Operational support; high-frequency tasks	Coordinate schedules, assist customers & consultants, view statistics
Manager	Strategic oversight; low-frequency usage	Manage staff accounts, approve content, access system-level reports
Admin	Top-level management	Global CRUD on all data, audit-log access, system settings

Role-based access control is enforced via `authenticateToken` and `authorizeRoles` middleware on the backend.

3 FUNCTIONAL REQUIREMENTS

3.1 Use Case Diagram

The comprehensive functionality of the GenCare Healthcare Platform system is illustrated in Figure 6. This use case diagram demonstrates the various interactions between different user roles and the system's core functionalities.

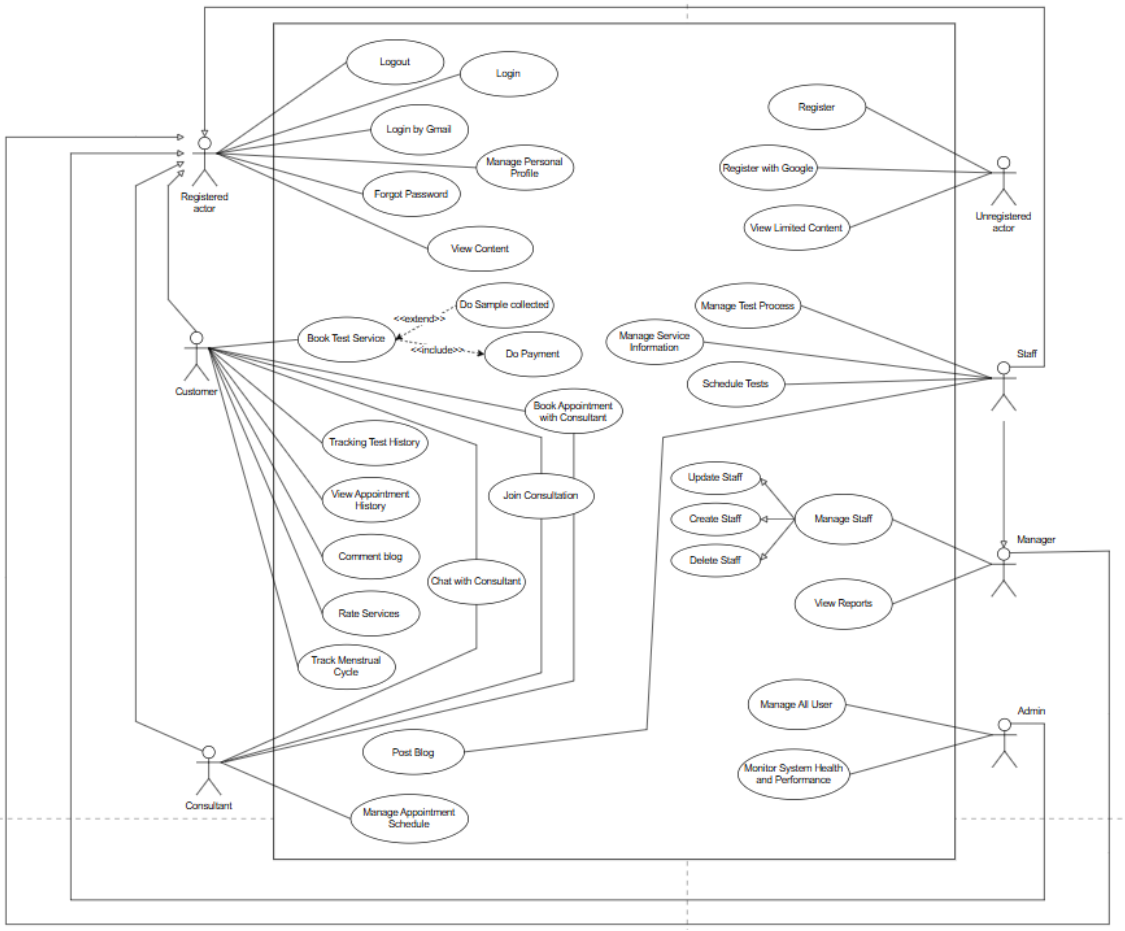

Figure 6: GenCare Healthcare Platform Use Case Diagram

The use case diagram shown in Figure 6 illustrates the comprehensive functionality of the GenCare Healthcare Platform, showing interactions between different user roles:

- **Unregistered Actor:** Can register, register with Google, and view limited content
- **Registered Actor:** Has access to login, logout, profile management, content viewing, and various healthcare services
- **Customer:** Can book test services, make payments, track test history, view appointment history, book appointments with consultants, join consultations, comment on blogs, rate services, and track menstrual cycles
- **Staff:** Manages test processes, service information, and schedules tests
- **Manager:** Handles staff management (create, update, delete staff) and views reports
- **Consultant:** Posts blogs and manages appointment schedules
- **Admin:** Manages all users and monitors system health and performance

3.2 Use Case Specifications

3.2.1 UC001 - Booking Appointment

Table 2: Booking Appointment Specification

Field	Description
ID and Name	UC-01 Booking Appointment
Created By	Group 3
Date Created	18/06/25
Primary Actor	Customer
Secondary Actors	Consultant
Description	Customer books a consultation appointment with a consultant. The system checks availability, validates business rules, and processes the booking. The appointment is created with pending status awaiting confirmation from the consultant.
Trigger	Customer indicates they want to book an appointment with a specific consultant by clicking booking appointment button.
Preconditions	PRE-1: Customer's identity has been authenticated and logged into the system PRE-2: Customer has a valid account with 'customer' role PRE-3: Customer has no other pending appointments PRE-4: Selected consultant exists in the system
Postconditions	POST-1: Appointment is created with 'pending' status in the system POST-2: Appointment information is recorded in the database POST-3: Appointment history is logged with creation details POST-4: Consultant can view the new appointment in their appointment list POST-5: Customer receives confirmation notification of successful booking
Normal Flow	**3.0 Book Appointment from Consultation System** 1. Customer selects desired consultant 2. System checks customer has no existing pending appointments 3. System validates appointment time must be at least 2 hours in advance 4. System verifies consultant exists and available 5. System retrieves consultant's weekly schedule for the selected date 6. System show consultant's available slots on the selected week 7. System checks appointment time is within working hours and not during break time 8. System creates new appointment with 'pending' status 9. System logs appointment creation in history audit trail 10. System returns success response with appointment details

Field	Description
Alternative Flows	**3.1 Book Appointment from Consultation without login** 1. Before step 1, if customer did not login, navigate to the login page. 2. After login successfully, navigate to step 1 and continue similar to the Normal Flow.
Exceptions	**3.E1 Customer Has Existing Pending Appointment** 1. At step 3, system detects customer already has a pending appointment 2. System displays error message "You already have a pending appointment. Please wait for confirmation or cancel the existing one before booking new appointment." 3. System suggests customer cancel existing appointment or wait for confirmation 4. Use case terminates **3.E2 Insufficient Lead Time** 1. At step 4, system calculates time difference between current time and appointment time 2. If time difference < 2 hours, system returns error with detailed time difference information 3. System suggests customer select alternative time slots 4. Use case terminates **3.E3 Authentication System Unavailable** 1. System detects customer is not authenticated or lacks 'customer' role 2. System returns 401/403 authentication error 3. System redirects to login page and terminates use case **3.E4 Consultant Database Connection Lost** 1. At step 5, system cannot find consultant with provided ID due to database issues 2. System returns error "Consultant not found" 3. System suggests trying again later and terminates use case
Priority	High
Frequency of Use	Approximately 30-60 times per day during business hours
Business Rules	BR-19: Customer can only have 1 pending appointment at a time BR-20: Appointment must be booked at least 2 hours in advance BR-21: Appointment duration minimum 15 minutes, maximum 1 hours BR-22: No overlapping appointments allowed for the same consultant BR-23: Appointments cannot be scheduled during consultant's break time

Field	Description
Other Information	The system must handle timezone processing for online consultations. Integration with weekly schedule system is required for availability verification. Appointment history logging must comply with audit requirements for legal traceability.
Assumptions	Consultants have properly configured weekly schedules. Customers have complete and valid profile information. Stable network connectivity between services.

3.2.2 UC002 - Booking STI Test

The Table 3 illustrates the Booking STI Test process - one of the main flows of our topic. It has all of the fields that is mentioned in the use case specification. Our flow starts with the selection of a specific STI test package from the service list, combined with the assumption that the user has already been authenticated, appropriately authorized as a customer, and that both the STI package and the corresponding test exist within the system. Normally, the service proceeds through the normal flow. However, if the user is not logged in, the process follows an alternative flow. In addition, there are three exceptions, which is described in the field of Exceptions. After finishing the booking, the order will be created with "booked" status and recorded in the database, then write details log for the traceability and accountability.

Table 3: Booking STI Test Specification

Field	Description
ID and Name	UC-02 - Book STI Test
Created By	Group 3
Date Created	18/06/25
Primary Actor	Customer
Secondary Actors	Staff, Consultant
Description	This use case describes the end-to-end flow of STI test booking and execution. After logging in, the customer selects a test date, adds notes, and submits the booking. The system validates the input, creates the order, updates the schedule, logs the action, and sends a confirmation email. On the appointment date, the customer visits the clinic, pays, and the consultant assigns test packages. The staff collects specimens, updates order status through testing phases, and inputs results. The consultant verifies the results, and the system emails the final report to the customer, completing the process.
Trigger	The customer clicks on booking STI test button from the service list.

Field	Description
Preconditions	PRE-1: The customer's identity has been authenticated and logged in the system PRE-2: Customer has a valid account with 'customer' role PRE-3: STI packages and tests exist in the system.
Postconditions	POST-1: The STI order is successfully stored and updated in the system. POST-2: Final test results are delivered to the customer. POST-3: Activity logs are recorded for audit and traceability.
Normal Flow	**3.0 Book STI Testing Service** 1. Customer logs in and is authenticated by the system. 2. Customer navigates to the STI Booking page. 3. Customer selects a test date and adds optional notes. 4. Customer confirms and submits the booking. 5. System creates the order, updates the test schedule, logs the action, and sends a confirmation email. 6. On the booking date, customer visits the hospital. 7. System updates order status to "Accepted", assigns a consultant, and selects test packages. 8. Customer makes payment. 9. Staff collects specimen and system updates status to "Specimen Collected" and then "Testing". 10. Staff inputs test results. 11. Consultant reviews and confirms the results. 12. System sends final results to the customer by email.
Alternative Flows	**3.1 Book STI Testing Service without login** 1. Before step 1, if customer did not login, navigate to the login page. 2. After login successfully, navigate to step 1 and continue similar to the Normal Flow.

Field	Description
Exceptions	**3.E1 Booking Submission Failed** 1. At Step 5, the system encounters a server or database error during order creation. 2. System displays an error: "Failed to create booking. Please try again later." 3. No order is stored in the system. 4. Use case terminates. **3.E2 Invalid or Past Order Date** 1. At Step 6, customer does not complete payment at the clinic. 2. System keeps the order in "Pending" payment status. 3. No specimen is collected, and test process does not proceed. 4. Use case pauses until payment is made. **3.E3 Authentication Failed** 1. Before Step 1, system detects that the customer is not authenticated or lacks the 'customer' role. 2. System returns 401 Unauthorized or 403 Forbidden error. 3. System redirects the customer to the login page. 4. Use case terminates.
Priority	High
Frequency of Use	Approximately 20-60 times per day during business hours

Field	Description
Business Rules	BR-28: Order dates cannot be in the past and must be valid dates. BR-29: Order notes are limited to 500 characters maximum. BR-30: Each test schedule is limited to maximum 10 orders per day. When limit is reached, the schedule becomes locked. BR-31: Locked schedules cannot accept new orders or modifications. BR-32: Appointment statuses must follow the defined state flow: • Booked ⇒ (Accepted, Canceled) • Accepted ⇒ (Processing, Canceled) • Processing ⇒ SpecimenCollected ⇒ Testing ⇒ Completed • Canceled and Completed are final states BR-33: All actions of status change must be logged for legal traceability. BR-34: Canceled orders cannot be reused in any case. BR-35: Customers can only cancel orders in 'Booked' status. BR-36: Staff/Admin/Consultant cannot directly set status to 'Processing' - this happens automatically when payment is confirmed for 'Accepted' orders. BR-37: Consultants must have 'Sexual Health' specialization to be assigned to STI orders. BR-38: Order date changes require schedule management - decreasing count from old schedule and increasing count for new schedule. BR-39: When payment is confirmed (is_paid = true) for an 'Accepted' order, the status automatically changes to 'Processing'. BR-40: User role authorization for order operations: • Customer: Can cancel booked orders, update order date • Staff/Admin: Can update payment status, assign consultants, update order date • Consultant: Can update test items and package items BR-41: Orders must be associated with a valid customer, schedule, and have a non-negative total amount. BR-42: Each test must be linked to an identified patient and their data must be kept confidential (Law 15/2023/QH15, Vietnam). BR-43: Tests require a valid doctor order or verified self-registration (Circular 49/2018/TT-BYT). BR-44: Test results must be signed by a certified consultant. BR-45: Admin actions must be logged for legal traceability and auditing.

Field	Description
Other Information	The system must handle timezone processing for online consultations. Integration with test schedule system is required for availability verification. STI Test history logging must comply with audit requirements for legal traceability.
Assumptions	Customers have valid and authenticated accounts. Available time slots are up-to-date. Logging service is always operational.

3.2.3 UC003 - Manage Menstrual Cycle

Table 4 presents the Menstrual Cycle Management use case, one of the core functionalities of our system. This use case supports customers in tracking their menstrual cycles by selecting period days via a calendar interface. The process assumes that the customer is already authenticated and authorized as a valid system user, with access to the Menstrual Tracking page. Once period dates are selected and submitted, the system processes the input to compute ovulation dates and fertile windows, stores the new data, clears previous cycle information, and logs the operation for accountability. If the customer is not logged in, an alternative flow redirects them to the login page. The system also handles exceptional scenarios such as invalid inputs or failed authentication.

Table 4: Manage Menstrual Cycle Specification

Field	Description
ID and Name	UC-03 - Manage Menstrual Cycle
Created By	Group 3
Date Created	18/06/25
Primary Actor	Customer
Secondary Actors	-
Description	This use case allows a customer to track their menstrual cycle by selecting one or more period dates on a calendar. The system processes the input to generate cycle information, including ovulation date and fertile window, and stores the data for menstrual tracking.
Trigger	The customer clicks for choosing period days from the service list.
Preconditions	PRE-1: The customer's identity has been authenticated and logged in the system PRE-2: The user has access to the Menstrual Page.
Postconditions	POST-1: Menstrual cycle data is successfully stored in the system. POST-2: Calculated ovulation dates and fertile windows are available for future reference. POST-3: Previous cycle data is cleared and replaced with the updated one.

Field	Description
Normal Flow	**3.0 Track Menstrual Cycle** 1. Customer logs in and is authenticated by the system. 2. Customer navigates to the Menstrual Page. 3. System displays a calendar interface. 4. Customer selects one or more period dates and optionally adds notes. 5. Customer clicks Save to submit the data. 6. System validates input, merges dates into cycles, and calculates ovulation and fertile window. 7. System clears previous cycles, creates new ones, and shows a success notification.
Alternative Flows	**3.1 Manage Menstrual Cycle Service without login** 1. Before step 1, if customer did not login, navigate to the login page. 2. After login successfully, navigate to step 1 and continue similar to the Normal Flow.
Exceptions	**3.E1 Invalid Period Dates** If the selected dates are invalid (e.g., duplicates, format errors): 1. The system shows an error message: "Invalid date(s). Please check and try again." 2. The user must reselect the dates. 3. Returns to step 4. **3.E2 Authentication Failed** 1. Before Step 1, system detects that the customer is not authenticated or lacks the 'customer' role. 2. System returns 401 Unauthorized or 403 Forbidden error. 3. System redirects the customer to the login page. 4. Use case terminates.
Priority	High
Frequency of Use	Everyday

Field	Description
Business Rules	BR-46: Period days data cannot be empty and must be provided as valid date arrays. BR-47: Cycle length must be between 21-35 days to be considered regular. BR-48: Period length must be between 2-7 days to be considered regular. BR-49: Cycle length is calculated as days between consecutive period start dates. BR-50: The last cycle in sequence has cycle length set to 0 (incomplete cycle). BR-51: Ovulation date is predicted as 14 days before next cycle start for regular cycles. BR-52: Fertile window spans from 5 days before ovulation to 1 day after ovulation. BR-53: Only cycles with length 21-35 days get fertility predictions calculated. BR-54: Minimum 3 periods required to determine period regularity status. BR-55: Period days appearing across multiple cycles are considered duplicates. BR-56: Pregnancy chance is classified as 'high' on ovulation day, 'medium' during fertile window, 'low' otherwise.
Other Information	The system must handle timezone processing for true time recommendation. Integration with test schedule system is required for availability verification.
Assumptions	The user is authenticated with a valid customer account and has access to the Menstrual Cycle Tracking page. It is assumed that the user enters accurate period dates. The system correctly handles time zones for calculating ovulation and the fertile window. The backend services for data processing, storage, and logging are available and functional. Additionally, the rules for identifying menstrual cycles are predefined and consistent throughout the use case.

3.3 State Diagrams

3.3.1 STI Test Order State Diagram

The STI Test Order State Diagram, presented in Figure 7, illustrates the life cycle of STI test booking within the system. The diagram depicts six distinct states and their corresponding transitions through a unidirectional flow model. As shown in Figure 7, the system begins in the "Available" state, representing time slots ready for customer booking, and transitions to Booked upon successful reservation. From this central state, three potential pathways emerge based on external events: the system may transition to "No-show" when scheduled time expires without customer attendance, to "Canceled" if customers proactively cancel their appointments, or to "In progress" when customers arrive and consultation commences,

ultimately reaching "Completed" upon service conclusion. This state machine model is designed to ensure effective handling of all scenarios that may arise during the STI test booking process. The Booked state serves as the central point for determining the next steps, ensuring accurate tracking and analysis of appointment outcomes.

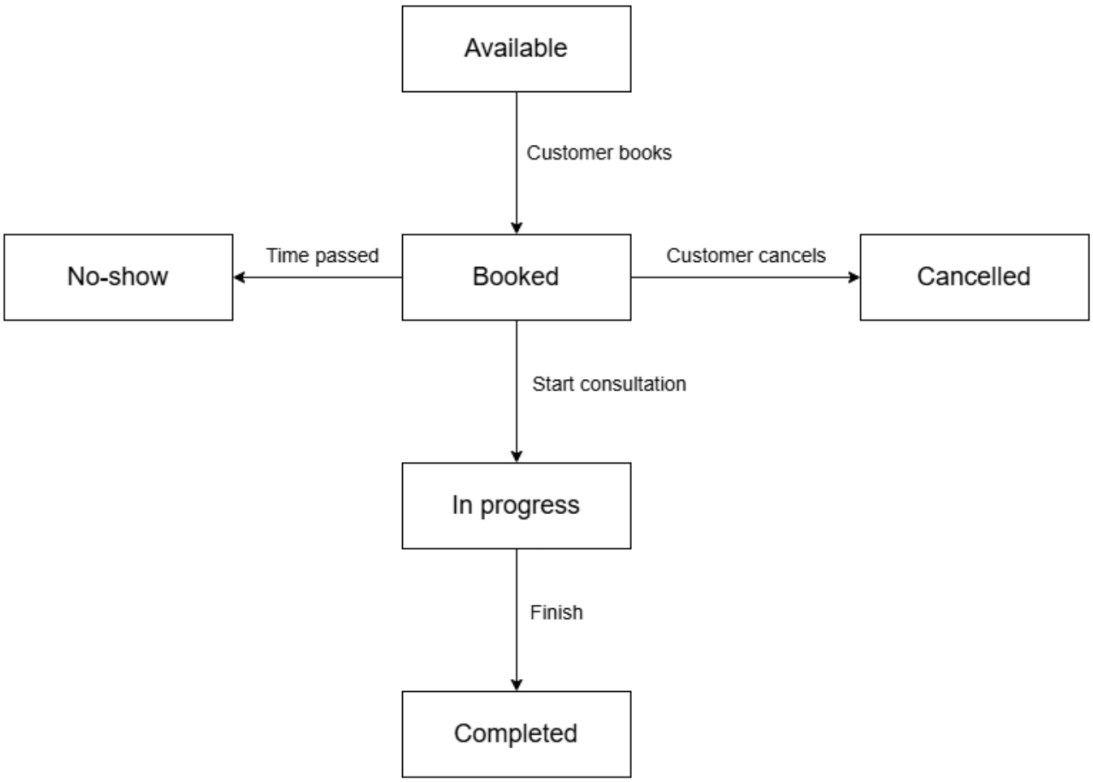

Figure 7: Booking STI Test State Transition Diagram

3.3.2 Result Test State Diagram

The STI test result processing state diagram, illustrated in Figure 8, shows the comprehensive process of handling STI test results from sample collection to final result delivery. The diagram demonstrates eight main states with one branch for handling rejected samples, following a clear and step-by-step flow with built-in quality checks. As depicted in Figure 8, the process starts with "Sample collected", then moves through transportation to the laboratory. At "Received by lab", the system performs sample quality validation. If a sample is not of adequate quality, it is directed to "Sample-rejected" to ensure result accuracy. Acceptable samples continue through "In analysis" and "Analysis complete", followed by a review phase with "Pending review" and "Reviewed", where a qualified healthcare professional checks and approves the results—a critical step for STI testing due to its medical significance and emotional impact. The process concludes with "Reported" for documentation and "Delivered", when results are communicated to the patient. This comprehensive flow ensures accuracy, efficiency, and proper medical review, with appropriate handling of rejected samples.

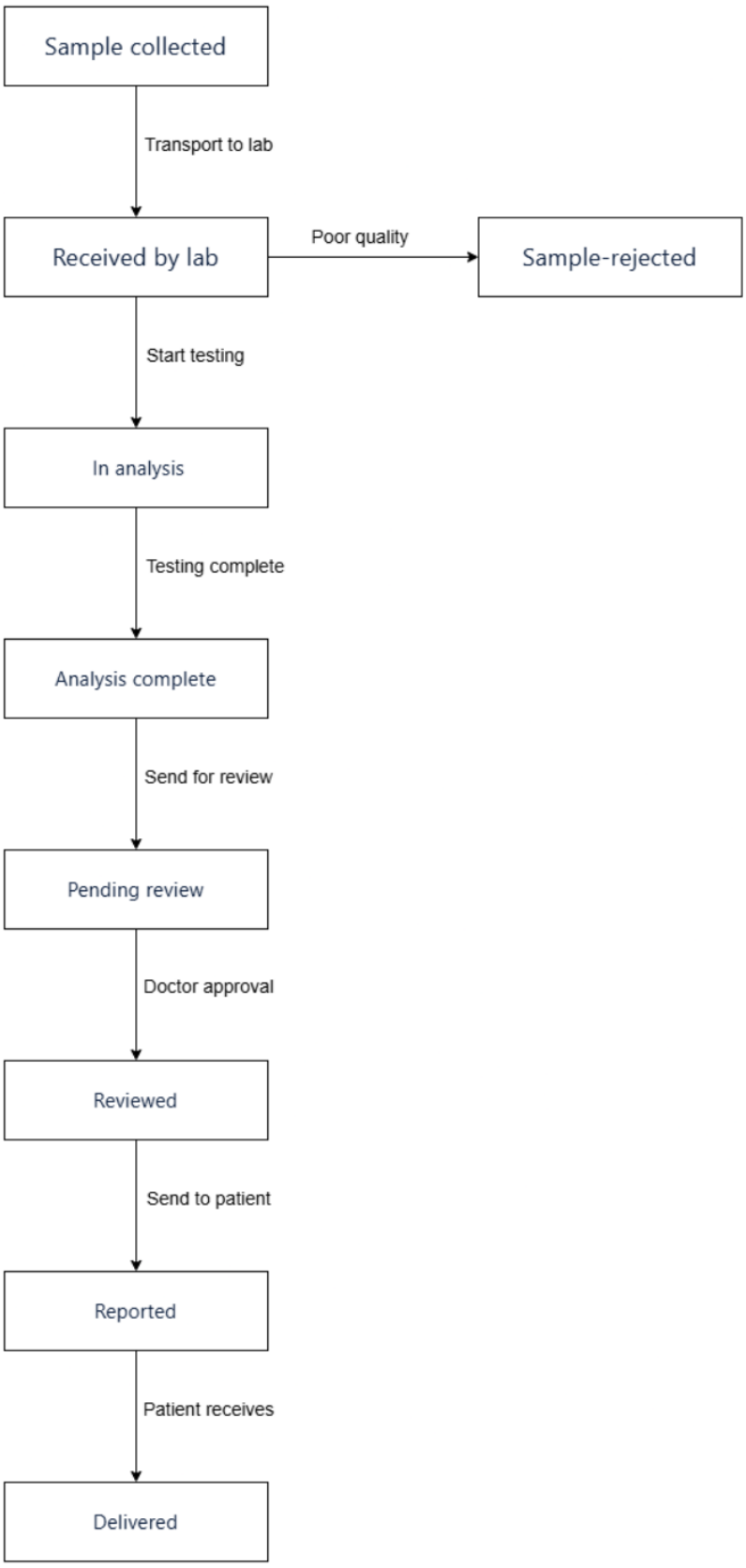

Figure 8: STI Test Result Processing State Transition Diagram

3.4 Data Flow Diagrams

3.4.1 Level 1 DFD - Booking Appointment

The Level-1 Data Flow Diagram for "Booking Appointment" (Figure 9) decomposes the top-level booking process into five distinct subprocesses connected to three internal data stores and two external entities. The primary external actors are the Customer, who initiates all booking interactions, and the Consultant, who ultimately receives the confirmed appointment.

The workflow begins with Process 1 "Validate Request," which authenticates the customer against the Customer Database and either accepts or rejects the booking access request. If validation succeeds, Process 2 "Select Consultant" retrieves a filtered consultant list from the Consultant Database in response to the customer's search criteria. Next, Process 3 "Select Schedule" queries the Weekly Schedule Database to return available time slots for the chosen consultant.

After the customer selects a slot, Process 4 "Book Appointment" writes a new Appointment Record to the Appointment Database and simultaneously creates an audit entry in the History Database. Finally, Process 5 "Confirm/Cancel" allows the customer to confirm or cancel the booking; the system updates appointment status in the Appointment Database, appends a history record, and, if confirmed, forwards the appointment details to the Consultant, who updates his or her weekly schedule accordingly.

As shown in Figure 9, data flows illustrate authentication responses, consultant lists, available times, booking commands, appointment records, and status notifications, ensuring that (a) only authenticated customers can book, (b) consultant schedules remain consistent, and (c) every action is permanently logged for traceability.

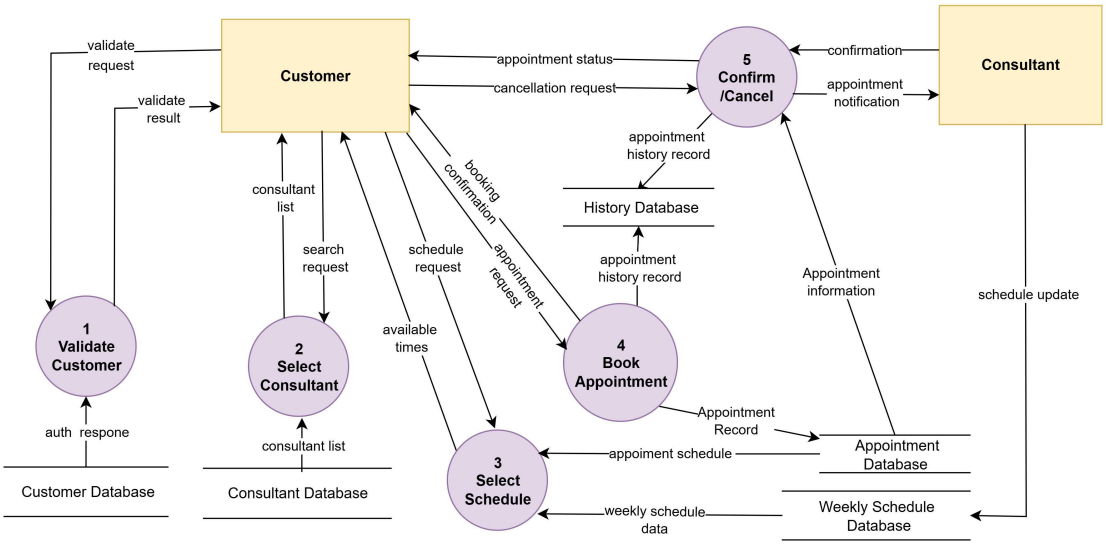

Figure 9: Booking Appointment Data Flow Diagram Level 1

3.4.2 Level 1 DFD - Booking STI

The main flow "Booking STI" contains two parts. The first one is that Customers do the date booking on the website, and then meeting consultant for package/tests consultation. The second one is that receiving samples from Customer and do the testing in the laboratory, then conducting diagnosis and send the test result to the customer.

The Level 1 Data Flow Diagram for STI test booking and consultation, as shown in Figure 10, illustrating the complete information flow throughout the appointment booking

and consultation process, highlighting the interactions between customers, staff, consultants, and the system's internal processing steps with associated databases. Its workflow begins with customer authentication to ensure secure access, followed by appointment selection based on real-time availability. Upon confirmation, a booking order is created and logged for traceability. The customer then selects a consultant, whose details are forwarded to staff for coordination. Finally, the consultant recommends appropriate test packages, drawing from relevant STI test and package databases to tailor options to the customer's needs.

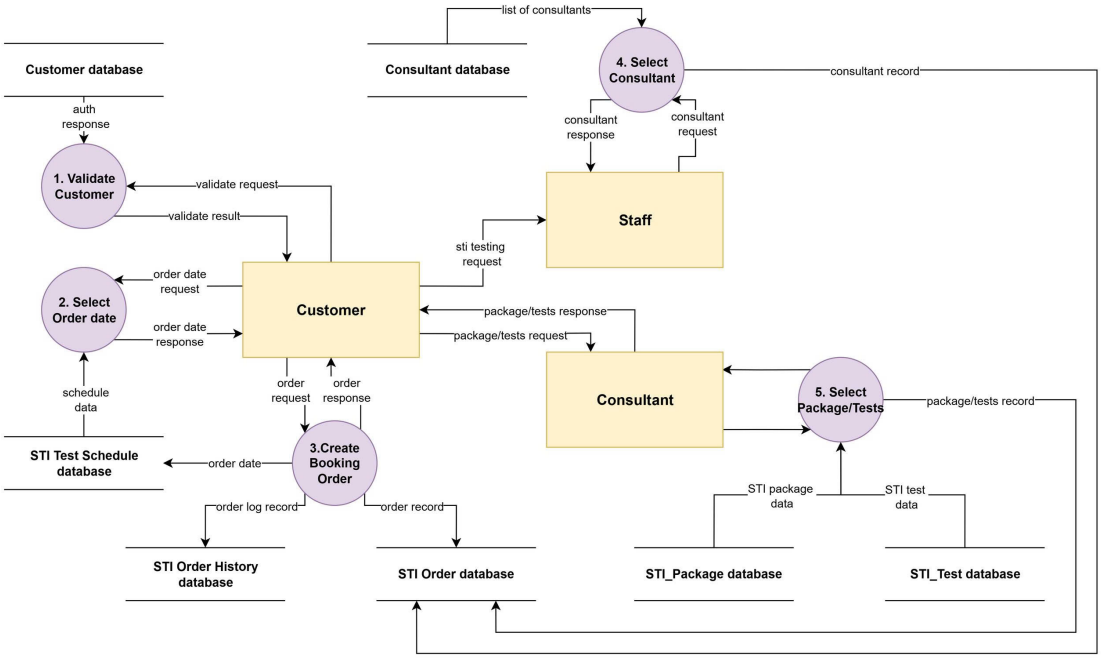

Figure 10: Booking STI Data Flow Diagram Level 1

The Level 1 Data Flow Diagram for order processing and result management (Figure 11) demonstrates the information flow from payment processing through final result delivery, emphasizing the systematic handling of test execution and result validation. In the process 6, the Update Order Status process plays a critical role in enabling staff to effectively monitor and manage each order throughout the testing workflow. By maintaining up-to-date status information in the STI Order database, this process helps prevent task omissions and minimizes errors, ensuring that all procedural steps are systematically tracked and executed. The Update Payment Status process (Process 7) manages payment interactions between customers and staff, recording payment status within the STI Order database to support transparency and financial accountability. Diagnostic operations are conducted through the Testing and Create Result Items process (Process 8), which involves coordinated specimen collection and result generation. Test outcomes are securely stored in the STI Result database for structured management. Finally, the Confirm the Final Result process (Process 9) involves consultant to validate all test results as well as diagnosis making. This step ensures clinical accuracy and completeness before formally updating the order as completed and preparing the results for customer delivery. After confirmation, result will be sent to customer through email.

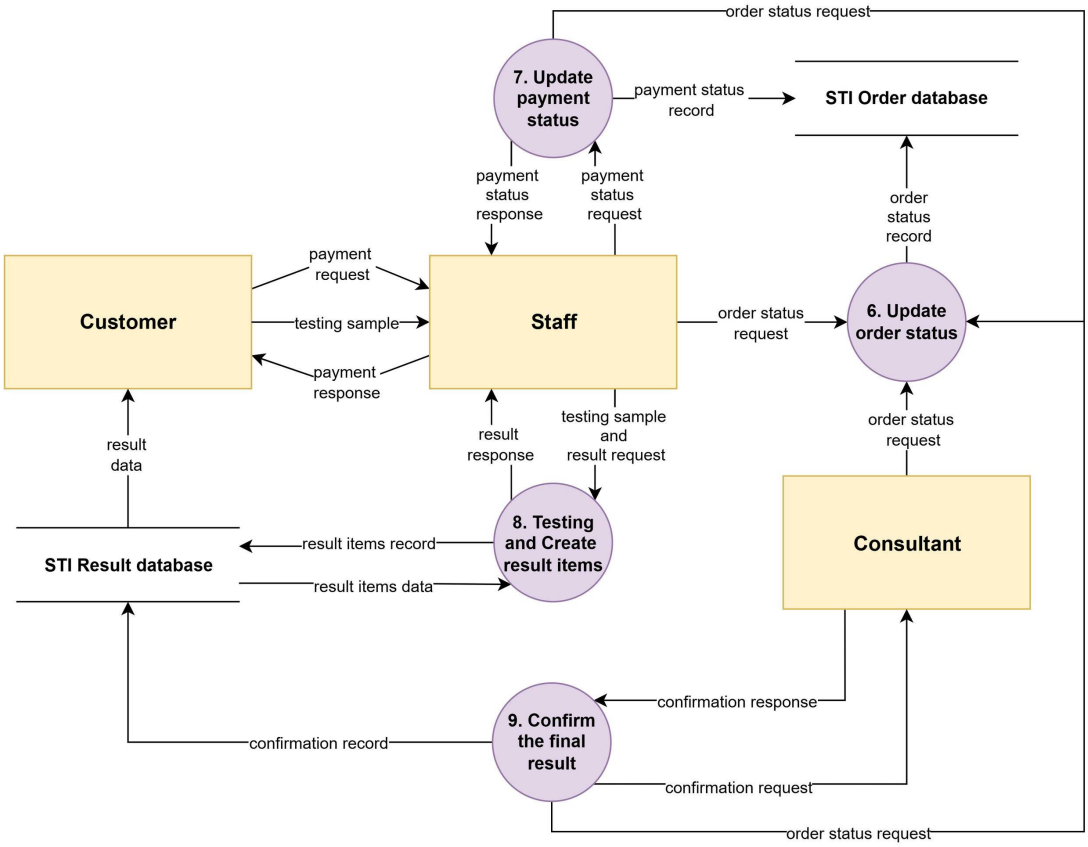

Figure 11: Process Order Status and Result Data Flow Diagram Level 1

3.5 Logical Data Model

The logical data model (Figure 12) defines the comprehensive data structure for the healthcare consultation and STI testing management system. The model comprises 17 interconnected entities organized into distinct functional domains as illustrated below:

User Management Domain: The system supports multiple user types through *User*, *Staff*, *Consultant*, and *Customer* entities. The *User* entity serves as the base table containing common authentication and profile information, while specialized entities inherit role-specific attributes. *Staff* manages system operations, *Consultant* provides medical consultations, and *Customer* represents service recipients.

Appointment Management Domain: The core scheduling functionality is implemented through *Appointment*, *AppointmentHistory*, *WeeklySchedule*, and *WorkingDay* entities. The *Appointment* entity manages booking requests and status tracking, while *AppointmentHistory* maintains a complete audit trail of all appointment modifications. *WeeklySchedule* and *WorkingDay* entities define consultant availability patterns and working hours.

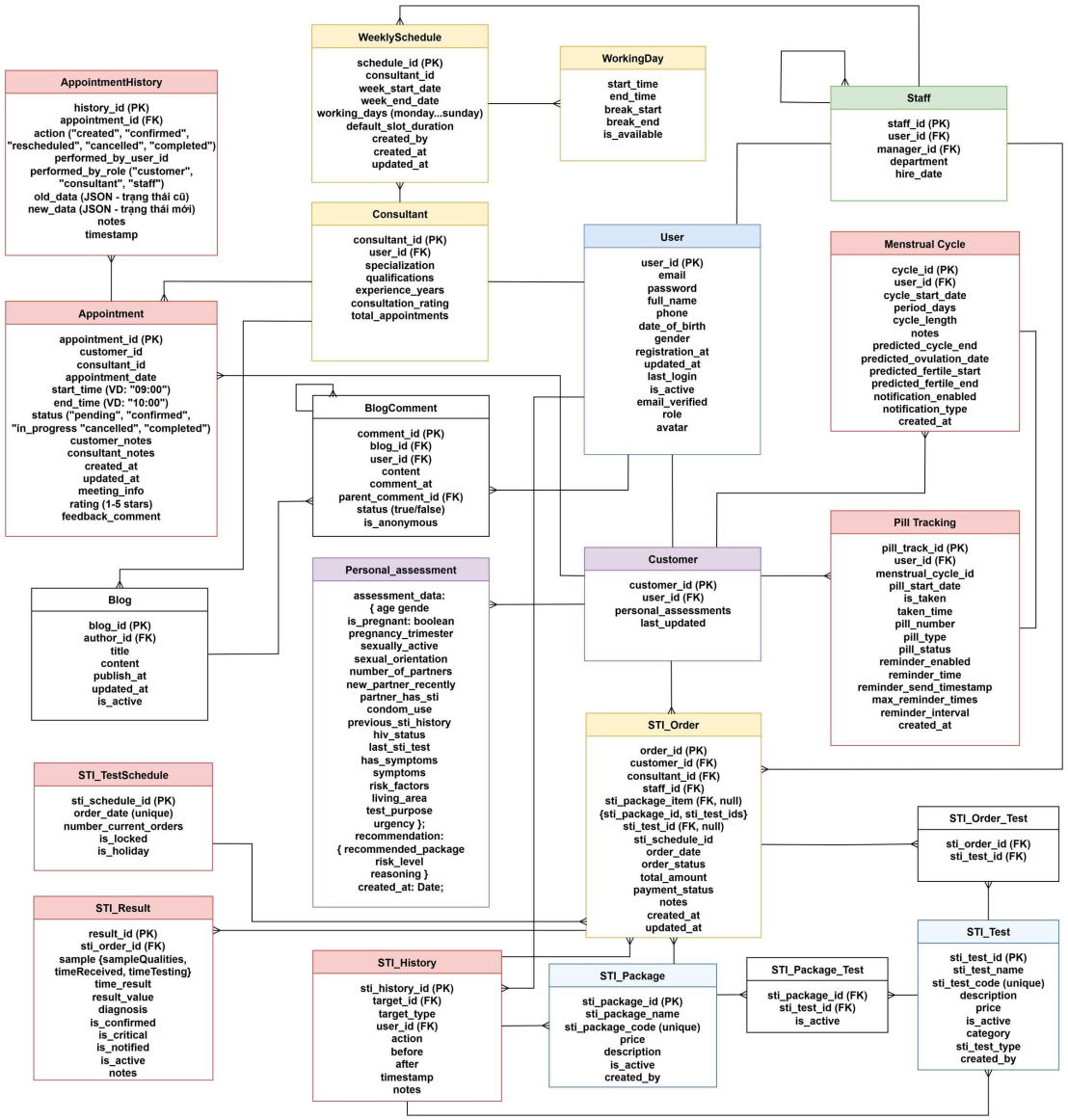

Figure 12: Logical Data Model for Healthcare Consultation & STI Testing System

Healthcare Tracking Domain: Specialized healthcare monitoring is supported through *MenstrualCycleEntries* and *PillTracking* entities. These entities enable personalized women's health monitoring, cycle prediction, and medication adherence tracking with automated reminder capabilities.

STI Testing Domain: The STI testing workflow encompasses *STI_Order*, *STI_Package*, *STI_Test*, *STI_Package_Test*, *STI_Order_Test*, and *STI_TestSchedule* entities. This domain supports flexible test ordering, package bundling, and scheduling. The many-to-many relationships between packages and tests allow for customizable test combinations.

Content Management Domain: Educational content and community engagement are facilitated through *Blog* and *BlogComment* entities, supporting knowledge sharing and user interaction within the healthcare community.

System Audit Domain: The *STI_AuditLog* entity provides comprehensive system activity logging for security monitoring, compliance tracking, and performance analysis.

All entities shown in Figure 12 implement standard temporal attributes (*created_at*, *updated_at*) to ensure data lineage and support regulatory compliance requirements. The model maintains referential integrity through properly defined foreign key relationships and supports both real-time transactional processing and historical data analysis.

4 NON-FUNCTIONAL REQUIREMENTS

4.1 Usability

4.1.1 U-1 — User Interface Design

The system shall provide an intuitive, responsive interface based on a Tailwind CSS design system and the shadcn/ui component library. Primary tasks shall be achievable within three clicks from the main dashboard and navigation shall remain consistent across all modules.

4.1.2 U-2 — Accessibility

The interface shall comply with WCAG 2.1 Level AA: support screen readers, full keyboard navigation and a high-contrast mode. Text shall scale up to 200 % without loss of functionality.

4.1.3 U-3 — Learning Curve

A new user shall be able to register and book a consultation within ten minutes without external training. Context-sensitive help shall be available on every screen.

4.2 Reliability

4.2.1 R-1 — System Availability

The platform shall maintain 99.9 % uptime during operational hours, equivalent to a maximum of 8.77 hours of unscheduled downtime per year. Planned maintenance will be scheduled from 02:00 to 06:00 local time with at least 48 hours' notice.

4.2.2 R-2 — Backup & Recovery

Incremental backups shall run every six hours, with full backups daily and a 30-day retention window. Recovery Point Objective (RPO) is one hour; Recovery Time Objective (RTO) is four hours.

4.2.3 R-3 — Error Handling

All critical errors shall be logged automatically and trigger immediate alerts to administrators. User-facing messages must be clear and actionable while revealing no sensitive information.

4.3 Performance

4.3.1 P-1 — Response Time

Under normal load, 95 % of page navigations must complete in $\leq$ 2 s; REST API calls $\leq$ 500 ms; complex database queries $\leq$ 2s.

4.3.2 P-2 — File Upload

The avatar-upload facility shall accept image files up to 5 MB and confirm upload within 30 s.

4.3.3 P-3 — Concurrent Usage

A single application node shall sustain at least 200 simultaneous sessions without perceptible slowdown. Horizontal scaling will support $\geq$ 1000 concurrent active users and up to 5000 at peak when auto-scaling is enabled.

4.3.4 P-4 — Video Consultation Quality

Video sessions (powered by Google Meet) shall provide at least 720 p @ 30 fps video and ¡ 150 ms audio latency for up to 100 concurrent meetings; bandwidth optimization is delegated to the provider.

4.4 Reusability

4.4.1 Re-1 — Modular Service Layer

Business rules must be encapsulated in independent TypeScript service modules exposing clear interfaces so that controllers, background jobs or future micro-services can reuse them without modification.

4.4.2 Re-2 — Unified Front-End Components

All screens shall reuse or extend the shared Tailwind / shadcn/ui component set, preserving naming and styling conventions.

4.4.3 Re-3 — API Standardisation

All outward-facing endpoints shall conform to a common response envelope and error pattern. An OpenAPI 3.0 specification will be maintained as the single source of truth.

4.5 Scalability

4.5.1 S-1 — Elastic Application Layer

Because the application is stateless, additional instances can be added or removed behind a load balancer. Autoscaling rules: scale-out at 70 % CPU utilisation, scale-in at 30 %.

4.5.2 S-2 — Data-Store Growth

The current schema and indexing strategy shall support at least 100.000 customers and one million appointment records before sharding or read replicas become necessary.

4.5.3 S-3 — Test Automation Expansion

The Selenium + pytest automation suite shall run tests in parallel; additional browser nodes can be added incrementally to maintain execution time as coverage increases.

5 ALGORITHM

This section presents the core algorithms implemented in the GenCare Healthcare Platform. These algorithms handle critical business logic for personal STI screeing, menstrual cycle prediction, and pill tracking management.

5.1 Personal STI Screening Algorithm

The Personal STI Screening Algorithm implemented in the GenCare Healthcare Platform represents a comprehensive, evidence-based decision support system designed to provide personalized STI testing recommendations. The algorithm follows the CDC (Centers for Disease Control and Prevention) guidelines 14 and incorporates clinical best practices for sexual health screening.

The algorithm employs a hierarchical decision tree structure that prioritizes high-risk factors and systematically evaluates patient characteristics to determine appropriate testing packages. As illustrated in Figure 13, the algorithm processes assessment data through multiple decision points to generate tailored recommendations.

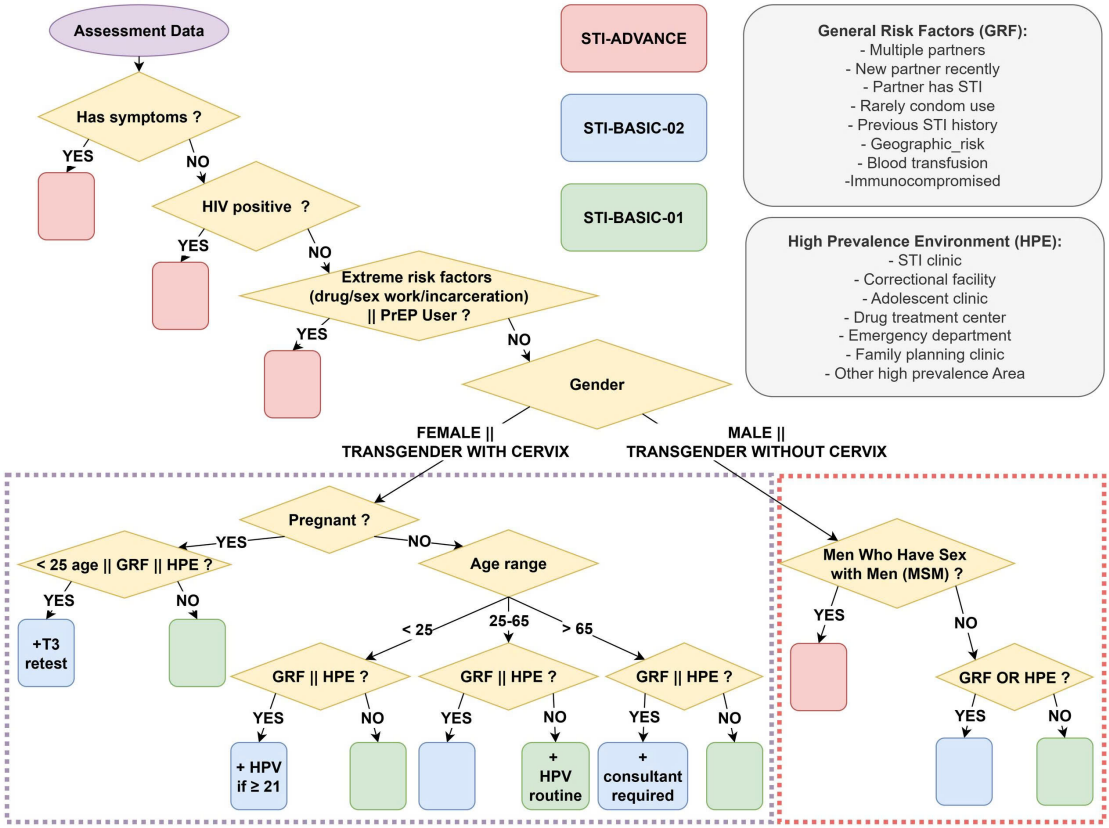

Figure 13: Personal STI Screening Algorithm Decision Tree

5.1.1 Tiered Risk Assessment Structure

The algorithm implements a five-tier prioritization system that ensures the most critical risk factors are evaluated first:

Tier 1: Acute Symptom Assessment

- Highest priority evaluation - first decision point in the flowchart
- Immediate recommendation for STI-ADVANCE package if symptoms present
- Bypasses all other assessment criteria when symptoms are detected
- Direct pathway to comprehensive testing regardless of other risk factors

Tier 2: HIV Status Evaluation

- Second critical checkpoint after symptom assessment
- HIV-positive individuals automatically receive STI-ADVANCE recommendation
- Critical for determining screening frequency and comprehensiveness
- Recognizes immunocompromised status requiring enhanced surveillance

Tier 3: Extreme Risk Factor Assessment

- Evaluates high-risk behaviors: injection drug use, sex work, incarceration
- Includes Pre-exposure Prophylaxis (PrEP) users as high-risk population
- Immediate escalation to STI-ADVANCE package for comprehensive testing
- Addresses populations with elevated STI transmission risk

Tier 4: Gender-Specific Population Identification

- Primary gender pathway differentiation point in the algorithm
- Female/Transgender with Cervix: Left pathway with pregnancy considerations
- Male/Transgender without Cervix: Right pathway with MSM-specific evaluation
- Men Who Have Sex with Men (MSM): Automatic STI-ADVANCE recommendation
- Gender-appropriate screening protocols based on anatomical considerations

Tier 5: Age and Risk-Stratified Assessment

- Age-stratified recommendations within gender-specific pathways
- General Risk Factors (GRF) and High Prevalence Environment (HPE) evaluation
- For females: Age groups <25, 25-65, >65 with different screening intensities
- For males: Risk-based screening with GRF/HPE considerations
- Pregnancy status: Special considerations with T3 retest requirements

5.2 Menstrual Cycle Algorithm

5.2.1 Rule for Grouping menstruation days

The rule for grouping menstruation days 15 is based on consecutive dates with a difference of exactly one day. Specifically, if two dates are consecutive, they are considered part of the same group. This grouping continues as long as each subsequent date is exactly one day after the previous one. A new group is initiated when the gap between two recorded menstruation dates exceeds one day.

5.2.2 Rule for Determining length of menstrual cycle

The menstrual cycle is the monthly series of changes a woman's body goes through in preparation for the possibility of pregnancy 16. It is required for the production of eggs and for the preparation of the uterus for pregnancy. The length of the menstrual cycle is calculated as the number of days between the start date of the current cycle and the start date of the previous cycle:

$$L_i = S_i - S_{i-1} \tag{1}$$

- L_i: Length of the i-th cycle
- S_i: Start date of the i-th cycle (present cycle)
- S_{i-1}: Start date of the $(i-1)$-th cycle (previous cycle)

5.2.3 Rule for Estimating Ovulation Date

Ovulation is the process during which an ovary releases an egg, typically occurring once in each menstrual cycle 16, 17. The ovulation date is the day in a woman's menstrual cycle when an ovary releases a mature egg.

In a regular cycle, ovulation generally occurs about 14 days before the start of the next period 17. Based on this principle, the ovulation date for the i-th cycle, denoted as O_i, can be estimated using the following formula:

$$O_i = S_i + (L_i - 14) \tag{2}$$

- O_i: Estimated ovulation date of the i-th cycle
- S_i: Start date of the i-th cycle
- L_i: Length of the i-th cycle (Equation 1)

5.2.4 Rule for Determining the Fertile Window

The fertile window refers to the range of days during a menstrual cycle when the chances of conception are highest. This period includes the days leading up to ovulation and shortly after it, as sperm can survive in the female reproductive tract for up to 5 days, and the egg remains viable for about 24 hours after ovulation 18. It is mentioned as the formula below, which captures the 6-day period when intercourse is most likely to result in pregnancy, beginning 5 days before ovulation and ending 1 day after.

$$F_i = [O_i - 5,\ O_i + 1] \tag{3}$$

- F_i: Fertile window of the i-th cycle (a range of days)
- O_i: Estimated ovulation date of the i-th cycle (Equation 2)

5.2.5 Rule for Determining Cycle Regularity

Normally, menstrual cycle is fluctuated about from 21 to 35 days per cycle 17, 18. However, there are some situations, which is out of the range. This is irregular menstrual cycle. And the cycle regularity helps assess whether a user's menstrual cycles fall within a typical, healthy range or not. It is denoted as R_{cycle}, is determined using the following conditions:

$$R_{cycle} = \begin{cases} \text{regular} & \text{if } \forall L_i \in [21, 35] \\ \text{irregular} & \text{if } \exists L_i \notin [21, 35] \\ \text{insufficient\_data} & \text{if } n < 3 \end{cases} \tag{4}$$

- R_{cycle}: Regularity status of the menstrual cycle
- L_i: Length of the i-th cycle (Equation 1)
- n: Number of cycles with available data

A cycle is considered regular if all recorded cycle lengths fall within the standard range of 21 to 35 days. If any cycle length falls outside this range, the pattern is labeled irregular. If fewer than 3 cycles are available (except for the recent cycle), the system reports insufficient data to assess regularity reliably.

5.2.6 Rule for Determining Period Regularity

Besides cycle regularity, the duration of menstruation (period) is also a very important factor in menstrual cycle management. The regularity of the period reflects the stability of hormonal activity and reproductive health. Consistently having periods that last within the medically accepted range of 2 to 7 days indicates a well-regulated cycle, while significant deviations may signal potential hormonal imbalances or underlying health issues. To systematically evaluate this aspect, we define a rule-based classification for period regularity using the following equation:

$$R_{period} = \begin{cases} \text{regular} & \text{if } \forall P_i \in [2, 7] \\ \text{irregular} & \text{if } \exists P_i \notin [2, 7] \\ \text{insufficient\_data} & \text{if } n < 3 \end{cases} \tag{5}$$

Here, R_{period} denotes the regularity status of menstruation duration, P_i represents the number of bleeding days in the i-th cycle, and n is the total number of recorded cycles. This classification aligns with medical standards is described in 16, which identify 2 to 7 days as the normal range for menstrual bleeding. If fewer than 3 cycles are available (except for the recent cycle), the system reports insufficient data to assess regularity reliably.

5.2.7 Rule for Determining Pregnancy Chance

A woman's likelihood of conceiving varies throughout her menstrual cycle. To assist with cycle tracking and fertility awareness, the following formula is used to estimate the chance

of conception on a given day D:

$$P_{\text{chance}} = \begin{cases} \text{high} & \text{if } D = O_i \\ \text{medium} & \text{if } D \in F_i \setminus \{O_i\} \\ \text{low} & \text{if } D \notin F_i \end{cases} \tag{6}$$

Where:

- P_{chance}: Probability of conception
- D: Current day
- O_i: Ovulation day (Equation 2)
- F_i: Fertile window (Equation 3)

The chance is high on the ovulation day (O_i), medium on other days within the fertile window (F_i), and low outside this window. This estimation helps users track their cycle and understand when they are most likely to conceive.

5.3 Pill Tracking Algorithm

5.3.1 Medicine Type Management

The system supports three common types of combination birth control pills, each with its own hormone schedule and cycle length:

- **21-day pills**: Contain 21 active (hormone) pills followed by 7 pill-free days. Users take a pill daily for 21 days, then pause for 7 days before starting a new pack.
- **24+4 pills**: Contain 24 active pills and 4 placebo (non-hormone) pills. Users take one pill daily for 28 days with no break between packs.
- **21+7 pills**: Include 21 active pills and 7 placebo pills. Similar to the 21-day type but with placebo pills provided to maintain daily pill-taking habit.

Each pill type is defined by two main properties:

- `totalDays`: Total number of days in the cycle (usually 28).
- `hormoneDays`: Number of days with active hormone intake.

To summarize, three types of pills are mentioned in the following formula:

$$\begin{aligned} \text{Config(pillType)} &= \{\text{totalDays} : T,\ \text{hormoneDays} : H\} \\ \text{where } (T, H) &= \begin{cases} (21,\ 21) & \text{if pillType} = \text{"21-day"} \\ (28,\ 24) & \text{if pillType} = \text{"24+4"} \\ (28,\ 21) & \text{if pillType} = \text{"21+7"} \end{cases} \end{aligned} \tag{7}$$

Table 5: Table of updating pill type

From type	To type	Agree or Disagree	Condition	Updating rule
21-day	21+7	Agree	No	Add placebo pills from day 22-28
21-day	24+4	Agree	No	Day $\leq$ 21: Add 3 hormone pills from 22-24, 4 placebo pills from 25-28 Day >21: Add remaining placebo only
24+4	21-day	Agree	Yes	Only Day $\leq$ 21: Deactivate all pills before 21 days
24+4	21+7	Agree with condition	Yes	Only Day $\leq$ 21: Update 22-28 pills to placebo pill
21+7	21-day	Agree	No	Deactivate all after-21 pills
21+7	24+4	Disagree	No	None

5.3.2 Rule of updating pill type

In pill tracking systems, users may switch between three pill regimens: 21-day, 21+7, and 24+4. Table 5 summarizes the allowed transitions between these types, the system's agreement status, applicable conditions, and the corresponding update rules.

Notably, transitions are generally permitted from 21-day to either 21+7 or 24+4 without conditions, while updates from 24+4 to other types require the user to be on or before Day 21. Any attempt to switch from 21+7 to 24+4 is rejected to maintain the integrity of the pill cycle structure. These rules help ensure logical consistency in pill tracking and preserve the expected flow of hormone and placebo phases.

5.4 Algorithm Complexity Analysis

5.4.1 Time Complexity Analysis

STI Screening Algorithm: $O(1)$ - Constant time complexity as it follows a deterministic decision tree with maximum depth of 5 tiers

Menstrual Cycle Prediction: $O(nlogn)$ - Linear time complexity where n represents the number of period days in input. The algorithm performs sequential operations: date normalization $O(n)$, consecutive day grouping $O(n)$ and then creating new cycle to database with $O(x \leq n)$. Additionally, TimSort is used for maintaining the period days input, with the time complexity of $O(nlogn)$.

Pill Schedule Generation: $O(n)$ - Linear complexity where n is the total pill days (21-28). The algorithm creates pill schedule objects through single iteration over total days, with database batch insertion of $O(n)$ for schedule creation and $O(1)$ for type switching operations.

5.4.2 Space Complexity Analysis

STI Screening Algorithm: $O(1)$ - Constant space for assessment data and reasoning array

Menstrual Cycle Prediction: $O(n)$ - The algorithm uses linear memory, mainly to store the input period days and their corresponding Date objects.

Pill Schedule Generation: $O(n)$ - Linear space to store pill schedule array where n represents total pill days (21-28), plus constant space for configuration objects and temporary variables.

6 SUPPORTING INFORMATION

The supporting information makes the SRS easier to use and provides additional context for implementation teams.

6.1 Testing Framework Implementation

Selenium Python Framework Architecture
The automated testing framework shall be implemented using Selenium WebDriver with Python, following the Page Object Model (POM) design pattern for maintainable and scalable test automation:

- **Framework Structure:** Organized into page objects, test cases, utilities, and configuration modules
- **Browser Support:** Chrome, Firefox, Safari, and Edge browsers with configurable driver management
- **Test Data Management:** External test data files in JSON and CSV formats for data-driven testing
- **Reporting:** Detailed HTML reports with screenshots for failed test cases and execution statistics
- **Parallel Execution:** Support for concurrent test execution using pytest-xdist for reduced testing time
- **CI/CD Integration:** Automated test execution in GitHub Actions and Jenkins pipelines

Test Coverage Requirements
The automated test suite shall achieve comprehensive coverage across all system functionality:

- **End-to-End Tests:** Complete user workflows including registration, consultation booking, and test ordering
- **UI Component Tests:** Validation of all user interface elements, forms, and interactive components
- **API Integration Tests:** Verification of all REST API endpoints with various input scenarios
- **Cross-Browser Tests:** Compatibility testing across supported browsers and devices
- **Performance Tests:** Load testing and response time validation for critical user paths

6.2 Technology Stack Documentation

Frontend Technology Stack

- React 18 + TypeScript for type-safe SPA development
- React Context API for client-side state management
- Tailwind CSS with Shadcn UI components for styling and design system
- Axios for HTTP communication with backend REST APIs
- Vite for fast local development and bundling

Backend Technology Stack

- Node.js 18 + Express.js with TypeScript for RESTful services
- MongoDB 5 with Mongoose ODM for data persistence
- Redis for caching, OTP storage and rate-limiting
- JSON Web Token (JWT) authentication with `bcrypt` password hashing
- Joi for request validation and data sanitisation
- Nodemailer for transactional e-mails (OTP, notifications)
- Passport Google OAuth 2.0 strategy for social login
- Nodemon + ts-node for developer hot-reload

Testing and DevOps Stack

- Jest + ts-jest for unit and integration testing of backend functions
- Selenium WebDriver (Python) for end-to-end UI test automation
- Postman for manual API testing and collections storage
- Docker + Docker Compose (Redis service) for local containerisation
- ESLint and Prettier for code quality and formatting