|  |  |  |
| --- | --- | --- |
| **Use case name:** | **UC01 - User Registration & Login** | |
| **Scenario:** | A new or returning user wants to access their Skinnovate dashboard. | |
| **Triggering event:** | User clicks “Register” or “Login” on the Skinnovate home page. | |
| **Brief description:** | Enables new users to create an account (via email/OTP or social login) and existing users to authenticate, then redirects them to their personalized dashboard. | |
| **Actors:** | * Patient * Dermatologist * Administrator | |
| **Related use cases:** | * UC02 – AI Skin Analysis (requires user to be authenticated) * UC03 – Appointment Booking & Management * UC04 – Dermatologist Review & Treatment Update | |
| **Stakeholders:** | * End users (patients, doctors, admins) * Clinic operations team | |
| **Preconditions:** | * System is online and accessible. * For registration: user email not already in database. * For login: user already has valid credentials. | |
| **Postconditions:** | * The user is logged in and gains access to their personalized dashboard. * Access control is applied based on the user’s role (e.g., patient, doctor, administrator). | |
| **Flow of activities:** | **Actor** | **System** |
| 1. Accesses Skinnovate portal 2. Selects Register 3. Enters registration details 4. — 5. Or selects Login 6. Enters credentials 7. — | Displays “Register” and “Login” options  Prompts for email, OTP/social login, and profile data  Validates inputs; if valid, creates new User record; sends confirmation  Redirects to dashboard  Prompts for email/password  Validates credentials  On success, redirects to dashboard |
| **Exception conditions:** | * **Invalid credentials:** display error “Email or password incorrect”—allow retry. * **Registration error:** for missing/invalid fields, highlight errors and block submission until corrected. | |

Figure 1: Fully developed use case description for ***User Registration & Login***

|  |  |  |
| --- | --- | --- |
| **Use case name:** | **UC02 –** **AI Skin Analysis** | |
| **Scenario:** | A logged-in patient wants a quick preliminary diagnosis of a skin condition using the AI engine. | |
| **Triggering event:** | Patient selects “AI Skin Analysis” from their dashboard. | |
| **Brief description:** | Patient uploads a skin image; the system checks quality, sends it to AI, then displays the AI’s diagnosis and confidence score. | |
| **Actors:** | * Primary: Patient * Secondary: AI System (automated) | |
| **Related use cases:** | * UC01 – User Registration & Login (must be logged in) * UC03 – Appointment Booking & Management (if AI suggests consultation) | |
| **Stakeholders:** | * Patients seeking fast insights * Dermatologists (for follow-up review) | |
| **Preconditions:** | * Patient is authenticated (UC01). * Device can capture/upload high-resolution images. | |
| **Postconditions:** | * A preliminary AI diagnosis (with confidence) is recorded. * Patient can choose to book a consultation based on results. | |
| **Flow of activities:** | **Actor** | **System** |
| 1. Clicks “AI Skin Analysis” 2. Uploads skin image 3. — 4. — 5. — 6. — | Prompts for image upload  Validates image quality  If valid, forwards image to AI engine  If poor, returns error “Please upload clearer image”  Runs AI analysis (≤ 5 sec) and returns diagnosis + confidence  Displays results with option “Book Consultation” if confidence < threshold |
| **Exception conditions:** | * **Poor image quality:** reject and prompt re-upload. * **Low AI confidence:** show “Low confidence—please book consultation.” | |

Figure 2: Fully developed use case description for ***AI Skin Analysis***

|  |  |  |
| --- | --- | --- |
| **Use case name:** | **UC03 – Appointment Booking & Management** | |
| **Scenario:** | A patient schedules, reschedules, or cancels a clinic appointment; admins monitor and adjust bookings. | |
| **Triggering event:** | Patient navigates to “Appointments” or admin opens the daily dashboard. | |
| **Brief description:** | Allows patients to view available slots, book/reschedule/cancel appointments, and triggers emergency handling when needed; administrators oversee and adjust schedules. | |
| **Actors:** | * Primary: Patient * Secondary: Clinic Administrator | |
| **Related use cases:** | * UC01 – User Registration & Login * UC02 – AI Skin Analysis (may prompt booking) * UC04 – Dermatologist Review & Treatment Update | |
| **Stakeholders:** | * Patients * Clinic staff (admins, doctors) | |
| **Preconditions:** | * User is authenticated (UC01). * Scheduling system is online and reflects real-time availability. | |
| **Postconditions:** | * Appointment is created, updated, or canceled. * Notifications sent to patient, admin, and dermatologist as appropriate. | |
| **Flow of activities:** | **Actor** | **System** |
| 1. Opens “Appointments” page 2. Selects desired slot 3. Confirms booking 4. Or selects “Reschedule” 5. Or selects “Cancel” 6. Admin views dashboard 7. Admin adjusts slot (if needed) | Displays calendar with available time slots  Checks slot availability  Creates Appointment record; sends confirmation notification  Prompts new slot selection; updates record; notifies parties  Marks appointment canceled; notifies parties  Displays all today’s appointments and statuses  Updates appointment records; notifies impacted patients |
| **Exception conditions:** | * **Slot already booked:** show “Selected slot unavailable—choose another.” * **Emergency booking:** if patient marks “Emergency,” triggers emergency workflow (reprioritize slots, bump non-urgent bookings, notify). | |

Figure 3: Fully developed use case description for ***Appointment Booking & Management***

|  |  |  |
| --- | --- | --- |
| **Use case name:** | **UC04 – Dermatologist Review & Treatment Update** | |
| **Scenario:** | A dermatologist examines AI-generated skin analysis results, confirms or modifies the diagnosis, and records a treatment plan. | |
| **Triggering event:** | An AI diagnosis report becomes available for a patient’s uploaded skin image (or a video consultation concludes). | |
| **Brief description:** | Enables the dermatologist to review AI suggestions, adjust the diagnosis if needed, and document a treatment plan in the patient’s record. | |
| **Actors:** | * Primary: Dermatologist * Secondary: AI System (automated process) | |
| **Related use cases:** | * UC02 – AI Skin Analysis * UC03 – Appointment Booking & Management | |
| **Stakeholders:** | * Patients (for accurate treatment) * Clinic staff (for scheduling follow-ups) * IT/QA (for ensuring AI accuracy) | |
| **Preconditions:** | * The patient’s AI analysis report exists and is accessible (UC02). * The dermatologist is authenticated and has appropriate role privileges. | |
| **Postconditions:** | * The finalized diagnosis and treatment plan are saved to the patient’s record. * Automated follow-up reminders are scheduled if required. | |
| **Flow of activities:** | **Actor** | **System** |
| 1. Opens patient’s AI report 2. Reviews diagnosis suggestion 3. Adjusts or confirms diagnosis 4. Selects or enters treatment plan details 5. Saves treatment plan | Retrieves AI suggestions and patient history  Displays confidence score and recommended options  Validates and logs the dermatologist’s decision  Presents treatment template and input fields  Persists data to EHR and schedules follow-up |
| **Exception conditions:** | * **AI report missing:** system displays “Report unavailable—please retry later. * **Validation error:** if required treatment fields are blank, system highlights missing items and blocks save. | |

Figure 4: Fully developed use case description for ***Dermatologist Review & Treatment Update***

|  |  |  |
| --- | --- | --- |
| **Use case name:** | **UC05 – E‑Prescription** | |
| **Scenario:** | After a dermatologist finalizes a treatment plan, the system generates an electronic prescription and sends it to the patient’s chosen pharmacy and insurer. | |
| **Triggering event:** | Dermatologist clicks “Issue e‑Prescription” at the end of a consultation. | |
| **Brief description:** | Allows a dermatologist to create, sign, and transmit a digital prescription; the system then forwards it to the pharmacy for fulfillment and to the insurance provider for claims processing. | |
| **Actors:** | * **Primary:** Dermatologist * **Secondary:** Pharmacy System, Insurance System | |
| **Related use cases:** | * UC04 – Dermatologist Review & Treatment Update * UC22 – Request E‑Prescription Refill | |
| **Stakeholders:** | * Patients (for timely medication access) * Partner pharmacies (for order fulfillment) * Insurance providers (for claim adjudication) | |
| **Preconditions:** | * The patient’s consultation record exists with a confirmed diagnosis and treatment plan. * Dermatologist is authenticated and authorized to prescribe. * Pharmacy and insurance endpoints are reachable. | |
| **Postconditions:** | * A signed e‑prescription is stored in the patient’s EHR. * Pharmacy receives the order and begins fulfillment. * Insurance provider receives claim data for coverage. | |
| **Flow of activities:** | **Actor** | **System** |
| 1. Clicks “Issue e‑Rx” Presents e‑prescription form 2. Enters medication, dosage, instructions Validates drug availability and dosing 3. Clicks “Sign & Send” 4. — 5. — 6. — 7. — | Presents e‑prescription form  Validates drug availability and dosing  Applies electronic signature  Transmits XML/HL7 message to Pharmacy System  Transmits claim data to Insurance System  Updates patient’s EHR with prescription record  Sends confirmation notice to dermatologist |
| **Exception conditions:** | * **\*\* formulary mismatch:\*\*** system displays “Selected medication not on formulary—choose alternative. * **\*\* transmission failure:\*\*** if pharmacy or insurer endpoint is down, system retries (up to 3×) and alerts IT on persistent errors. * **\*\* authorization denied:\*\*** insurance returns a denial code → system notifies dermatologist to adjust prescription. | |

Figure 5: Fully developed use case description for ***E‑Prescription***

|  |  |  |
| --- | --- | --- |
| **Use case name:** | **UC06 –** **System Recovery & Maintenance** | |
| **Scenario:** | IT staff detect or respond to system disruptions by running backups, restoring data, and ensuring overall platform health. | |
| **Triggering event:** | A scheduled maintenance window arrives or an alert reports a service outage. | |
| **Brief description:** | Enables IT/administrators to perform routine backups, archive data, restore from backups in case of failure, and monitor system health to meet availability targets. | |
| **Actors:** | * **Primary:** IT Support / System Administrator * **Secondary:** Backup Service, Monitoring Service | |
| **Related use cases:** | * UC41 – On Demand Data Backup * UC45 – Restore from Backup | |
| **Stakeholders:** | * All Skinnovate users (for uptime) * Compliance auditors (for data retention policies) | |
| **Preconditions:** | * Admin is authenticated with elevated privileges. * Backup storage and monitoring services are available. | |
| **Postconditions:** | * System data is safely backed up and archived. * Any failed components are restored to a healthy state. * System uptime SLA of 99.9% is maintained. | |
| **Flow of activities:** | **Actor** | **System** |
| 1. Logs into Admin Console 2. Triggers “Run Backup Now” 3. — 4. Receives alert of service failure 5. Confirms restore action 6. — 7. Opens Monitoring Dashboard 8. — | Displays Recovery & Maintenance options  Initiates snapshot to Backup Service  Moves older backups to cold storage (archival)  Presents “Restore from Backup” option  Coordinates with Backup Service to retrieve data  Restarts failed components; runs integrity checks  Shows system health metrics and alerts  Logs all maintenance tasks for audit |
| **Exception conditions:** | * **Backup failure:** system logs error, retries once, and emails IT if still failing. * **Restore integrity error:** if restored data fails checksum, system aborts and alerts senior admin. * **Monitoring service down:** system falls back to local logs and alerts on startup. | |

Figure 6: Fully developed use case description for ***System Recovery & Maintenance***

|  |  |  |
| --- | --- | --- |
| **Use case name:** | **UC08 – Two‐Factor Authentication Setup** | |
| **Scenario:** | A user enhances account security by enabling two‐factor authentication (2FA), choosing a delivery method, and verifying ownership. | |
| **Triggering event:** | User selects “Enable Two‐Factor Authentication” from their account security settings. | |
| **Brief description:** | Guides the user through selecting a 2FA method (SMS, email, or authenticator app), verifying the chosen channel, and activating 2FA for subsequent logins. | |
| **Actors:** | * Primary: User (Patient, Dermatologist, Administrator) * Secondary: Email/SMS Service | |
| **Related use cases:** | * UC01 – User Registration & Login * UC07a – Forgot Password | |
| **Stakeholders:** | * All users (for account security) * IT security team | |
| **Preconditions:** | * The user is authenticated and on their security settings page. * The user’s contact information (email or phone) is verified and up-to-date. | |
| **Postconditions:** | * Two‐factor authentication is enabled on the user’s account. * Future logins will require a second factor. | |
| **Flow of activities:** | **Actor** | **System** |
| 1. Navigates to Security Settings 2. Clicks “Enable Two‐Factor Authentication” 3. Selects preferred method (e.g., SMS) 4. Confirms contact info 5. Retrieves code (or scans QR in Auth App) 6. Enters setup code (or confirms app token) 7. Clicks “Activate 2FA” 8. — | Displays “Enable Two‐Factor Authentication” option  Presents choice of methods: SMS, Email, Auth App  Prompts for/validates phone number  Sends 2FA setup code or provisioning QR code  —  Verifies code/token  Updates security settings; marks 2FA as active  Displays “2FA Enabled” confirmation and backup codes |
| **Exception conditions:** | * **Invalid contact info:** “Phone number/email not recognized—please update profile.” * **Delivery failure:** on SMS/email bounce, system offers “Resend Code” and logs the failure. * **Invalid setup code**: “Code incorrect—try again” (up to 3 attempts), then abort back to method selection. * **Timeout:** if no code entered within 5 minutes, session expires and user must restart setup. | |

Figure 8: Fully developed use case description for ***Two‐Factor Authentication Setup***

|  |  |  |
| --- | --- | --- |
| **Use case name:** | **UC09 –** **Update User Profile** | |
| **Scenario:** | A user wishes to change their personal information such as name, phone number, contact method, or skin type preferences within their Skinnovate account. | |
| **Triggering event:** | The user navigates to the “Profile Settings” section and selects “Edit Profile. | |
| **Brief description:** | Allows users to securely update personal information stored in their account. The system validates and stores the changes, ensuring all linked records reflect the updated data. | |
| **Actors:** | * User (Patient, Dermatologist, Admin) | |
| **Related use cases:** | * UC01 – User Registration & Login * UC09 – Two Factor Authentication Setup * UC33 – Loyalty Points Award (some updates may affect preferences) | |
| **Stakeholders:** | * Users (for data accuracy and personalization) * System Admin (to ensure user info is current and secure) | |
| **Preconditions:** | * The user is authenticated and has access to their profile dashboard. * Profile data is available and loaded from the database. | |
| **Postconditions:** | * The user’s updated profile information is stored successfully. * A timestamped log of the changes is maintained. * A confirmation is shown to the user. | |
| **Flow of activities:** | **Actor** | **System** |
| 1. Navigates to Profile Settings 2. Clicks “Edit” 3. Enters updated data 4. Clicks “Save Changes” 5. – | Loads existing profile data into form fields  Enables editable fields  Validates fields (format, required info, email uniqueness)  Saves updates to database and creates audit log  Shows success message and updated profile view |
| **Exception conditions:** | * **Missing required fields:** System prompts user to fill in all required details. * **Invalid formats (e.g., email, phone):** System displays validation errors. * **Database update failure:** System shows an error and logs the issue. * **No internet/server timeout:** System informs the user to retry later. | |

Figure 9: Fully developed use case description for ***Update User Profile***

|  |  |  |
| --- | --- | --- |
| **Use case name:** | **UC010 –** **Track Skin Condition Progress** | |
| **Scenario:** | A patient logs recurring skin entries (photos and notes) to monitor their condition over time. | |
| **Triggering event:** | The patient selects “Track My Skin” from their dashboard. | |
| **Brief description:** | Enables the patient to upload periodic photos and text notes, view a timeline chart of improvements or flare‑ups, and set reminders for future check‑ins. | |
| **Actors:** | * **Primary:** Patient * **Secondary:** Dermatologist | |
| **Related use cases:** | * UC02 – AI Skin Analysis * UC12 – Manual Image Correction | |
| **Stakeholders:** | * Patients (for self‑monitoring) * Dermatologists (for remote progress review) | |
| **Preconditions:** | * The patient is logged in. * At least one prior skin entry exists (optional). | |
| **Postconditions:** | * A new skin entry (photo + notes) is saved. * Timeline chart updates to reflect the latest entry. * If configured, reminder for next entry is scheduled. | |
| **Flow of activities:** | **Actor** | **System** |
| 1. Clicks “Track My Skin” 2. Uploads new skin photo 3. Enters notes (e.g., date, symptoms) 4. Clicks “Save Entry” | Displays history timeline and “Add New Entry” form  Validates image format and size  Saves entry metadata and image to patient record  Updates timeline chart; persists reminder setting  Shows confirmation and next‑entry due date |
| **Exception conditions:** | * **Invalid image format:** prompt “Please upload JPG/PNG under 5 MB.” * **Missing note fields**: highlight empty fields and block save. * **Storage failure:** show “Unable to save entry—try again later.” | |

Figure 10: Fully developed use case description for ***Track Skin Condition Progress***

|  |  |  |
| --- | --- | --- |
| **Use case name:** | **UC011 – Manual Image Correction** | |
| **Scenario:** | When the system’s initial quality check flags a photo as inadequate, the patient crops, rotates, or retakes the image to meet analysis standards. | |
| **Triggering event:** | System returns a “Poor Image Quality” error after upload. | |
| **Brief description:** | Allows patients to correct or replace a photo that failed automatic validation, then resubmit for AI analysis. | |
| **Actors:** | * **Primary:** Patient | |
| **Related use cases:** | * UC02 – AI Skin Analysis * UC12 – Validate Image Quality | |
| **Stakeholders:** | * Patients (for accurate analysis) * System (to minimize false negatives) | |
| **Preconditions:** | * A recent image upload attempt was rejected by QC. | |
| **Postconditions:** | * The corrected image is accepted by the QC step. * The workflow returns to “Send to AI Engine.” | |
| **Flow of activities:** | **Actor** | **System** |
| 1. Sees “Poor Quality” alert 2. Edits image or selects “Retake” | Displays image correction tools (crop/rotate)  Validate corrected image size/resolution  If valid: proceeds to “Send to AI Engine”  If still invalid: shows same error and tools |
| **Exception conditions:** | * **Repeated failures:** after 3 attempts, offer “Contact Support.” * **User cancels correction:** abort flow and return to dashboard. | |

Figure 11: Fully developed use case description for ***Manual Image Correction***

|  |  |  |
| --- | --- | --- |
| **Use case name:** | **UC012 – Download Analysis Report** | |
| **Scenario:** | A patient or dermatologist wants an offline copy of the AI skin analysis results, including images, diagnosis summary, and confidence data. | |
| **Triggering event:** | User clicks “Download Report” on the analysis results page. | |
| **Brief description:** | Generates a PDF (or CSV bundle) of the AI’s findings and provides it for download or email delivery. | |
| **Actors:** | * **Primary:** Patient**,** Dermatologist | |
| **Related use cases:** | * UC02 – AI Skin Analysis * UC02d – View AI Confidence Trends | |
| **Stakeholders:** | * Patients (for personal records) * Dermatologists (for sharing with other providers) | |
| **Preconditions:** | * An AI analysis report exists for the selected image. * The user is authenticated. | |
| **Postconditions:** | * A report file (PDF/CSV) is generated. * Download is initiated or emailed as per user choice. | |
| **Flow of activities:** | **Actor** | **System** |
| 1. Clicks “Download Report” 2. Selects format and clicks “OK” 3. — 4. — 5. Sees “Download Complete” alert | Prompts format choice (PDF vs. CSV)  Gathers analysis data, images, and metadata  Renders report file  Initiates browser download or emails link to user  — |
| **Exception conditions:** | * **Report generation error:** show “Unable to generate report—please try again.” * **File size too large:** offer “Email link instead.” * **Email delivery failure:** notify user and log for support. | |

Figure 12: Fully developed use case description for ***Download Analysis Report***

|  |  |  |
| --- | --- | --- |
| **Use case name:** | **UC014 – Browse Dermatologists** | |
| **Scenario:** | A patient explores available dermatologists’ profiles to select one for consultation based on specialty, ratings, and availability. | |
| **Triggering event:** | The patient navigates to or clicks “Find a Dermatologist” from the dashboard. | |
| **Brief description:** | Enables patients to search, filter, and view dermatologists’ profiles—complete with specialties, years of experience, ratings, and available time slots. | |
| **Actors:** | * **Primary:** Patient * **Secondary:** Clinic Admin (for managing profiles) | |
| **Related use cases:** | * UC03 – Appointment Booking & Management * UC15 – Telederm Video Consultation | |
| **Stakeholders:** | * Patients (for choice of provider) * Clinic Administrators (for profile accuracy) * Dermatologists (for visibility) | |
| **Preconditions:** | * The patient is logged in. * Dermatologist profiles and availability data are up-to-date. | |
| **Postconditions:** | * The patient has identified one or more dermatologists to book. * The system logs the browsing activity for analytics. | |
| **Flow of activities:** | **Actor** | **System** |
| 1. Clicks “Find a Dermatologist” 2. Enters search or applies filters 3. — 4. Scrolls or refines filters 5. Clicks on a profile card 6. Chooses “Book Appointment” | Displays search/filter UI (specialty, rating, location)  Queries database for matching profiles  Presents list of dermatologists with summary cards  Updates list dynamically  Shows full dermatologist profile (bio, reviews, slots)  Transfers to Appointment Booking flow (UC03) |
| **Exception conditions:** | * **No matches found:** system displays “No dermatologists match your criteria.” * **Profile load failure:** shows “Unable to load profile—please retry.” | |

Figure 14: Fully developed use case description for ***Browse Dermatologists***

|  |  |  |
| --- | --- | --- |
| **Use case name:** | **UC015 – Telederm Video Consultation** | |
| **Scenario:** | A patient connects with a dermatologist via a secure video link for a remote consultation. | |
| **Triggering event:** | The patient clicks “Start Video Consultation” on a confirmed telederm appointment. | |
| **Brief description:** | Manages the end‑to‑end video session: launching the meeting room, handling connection issues, providing in‑session tools (screen share, image capture), and wrapping up with post‑consult notes. | |
| **Actors:** | * **Primary:** Patient, Dermatologist * **Secondary:** Video Service Provider | |
| **Related use cases:** | * UC03 – Appointment Booking & Management * UC04 – Dermatologist Review & Treatment Update | |
| **Stakeholders:** | * Patients (for remote care) * Dermatologists (for telehealth delivery) * IT Support (for video infrastructure) | |
| **Preconditions:** | * A telederm appointment exists and is scheduled for “now.” * Both patient and dermatologist have compatible devices and network connectivity. | |
| **Postconditions:** | * Session is logged (start/end times, participants). * Any captured screenshots or in‑session notes are saved to the patient’s record. * Follow‑up actions (e.g., prescriptions, referrals) are queued. | |
| **Flow of activities:** | **Actor** | **System** |
| 1. Clicks “Start Video Consultation” 2. — 3. Joins meeting 4. Both present 5. If network drop 6. — 7. Clicks “End Session” 8. — | Generates secure meeting link  Opens embedded video window  Connects both parties; displays “Waiting for other”  Enable in‑session tools: screen share, snapshot  System tries auto‑reconnect (3 attempts)  Alerts user on persistent failure  Closes video link; logs session metadata  Returns both to Skinnovate UI |
| **Exception conditions:** | * **Connection failure:** after retries, system prompts “Switch to audio only” or “Reschedule.” * **User drops session early:** system logs partial session and sends recap link. * **Tool error (e.g. screen share fail):** displays contextual error message and continues call. | |

Figure 15: Fully developed use case description for ***Telederm Video Consultation***

|  |  |  |
| --- | --- | --- |
| **Use case name:** | **UC16 – Appointment Check-In** | |
| **Scenario:** | A patient arrives at the clinic and checks in for their scheduled appointment. | |
| **Triggering event:** | Patient scans QR code or taps “Check In” from their appointment reminder. | |
| **Brief description:** | The system confirms the patient’s arrival, updates the appointment status to “In Progress,” notifies the dermatologist, and displays any pre-visit forms. | |
| **Actors:** | * **Primary: Patient** * **Secondary: ClinicAdmin (for manual overrides), ReceptionSystem** | |
| **Related use cases:** | * UC13 Appointment Booking & Management * UC15 Telederm Video Consultation | |
| **Stakeholders:** | * Patient (wants timely care) * Clinic staff (needs accurate arrival tracking) * Dermatologist (prepares for visit) | |
| **Preconditions:** | * Appointment exists and is in “Scheduled” status. * Patient has a valid appointment ID or QR check-in code. | |
| **Postconditions:** | * Appointment status = In Progress * Notification sent to dermatologist and waiting room display. * Any required pre-visit surveys presented to patient. | |
| **Flow of activities:** | **Actor** | **System** |
| 1. Tap “Check In” on mobile app 2. – 3. – 4. – 5. – 6. Complete pre-visit form(s) | 1. -  2. Validate appointment ID / QR code  3. Update appointment status → In Progress  4. Send notification to dermatologist  5. Display pre-visit form(s) to patient  6. -  7. Save form responses and confirm check-in |
| **Exception conditions:** | * **Invalid code/ID:** show “Invalid appointment” error and prompt retry. * **Already checked in**: inform patient “You have already checked in.” * **System offline:** display “Please wait, system unavailable” and queue check-in. | |

Figure 16: Fully developed use case description for ***Appointment Check-In***

|  |  |  |
| --- | --- | --- |
| **Use case name:** | **UC17 – Emergency Appointment Handling** | |
| **Scenario:** | A patient requests an urgent (“walk-in”) appointment outside normal scheduling. | |
| **Triggering event:** | Patient clicks “Emergency Appointment” or calls clinic emergency hotline. | |
| **Brief description:** | The system evaluates provider availability, suggests the next available slot (or places patient on emergency queue), confirms booking, and notifies clinic staff. | |
| **Actors:** | * **Primary: Patient** * **Secondary: ClinicAdmin, Dermatologist** | |
| **Related use cases:** | * UC13 Appointment Booking & Management * UC18 Automated Appointment Reminders | |
| **Stakeholders:** | * Patient (needs urgent care) * Clinic staff (manages urgent cases) * Dermatologist (responds to emergencies) | |
| **Preconditions:** | * Patient is registered and logged in * Emergency slots or queue defined in system | |
| **Postconditions:** | * Emergency appointment created (status = Scheduled or Queued) * Notifications sent to patient and clinic staff * Slot reserved or patient queued | |
| **Flow of activities:** | **Actor** | **System** |
| 1. Tap “Emergency Appointment” 2. Enter symptoms/details 3. – 4. – 5. – 6. Patient receives confirmation | 3. Check for free emergency slots  • If slot available → create appointment (Scheduled)  • Else → add to emergency queue  4. Send confirmation to patient  5. Notify ClinicAdmin & Dermatologist |
| **Exception conditions:** | * **Not logged in**: redirect to UC01 Login * **No emergency slots & queue full**: display “All urgent slots filled—please call clinic.” * **System error:** show “Unable to process emergency request, please call directly.” | |

Figure 17: Fully developed use case description for ***Emergency Appointment Handling***

|  |  |  |
| --- | --- | --- |
| **Use case name:** | **UC18 – Automated Appointment Reminders** | |
| **Scenario:** | The system automatically sends reminder notifications ahead of upcoming appointments. | |
| **Triggering event:** | Cron scheduler reaches reminder-lead time (e.g. 24 hrs before appointment). | |
| **Brief description:** | The system identifies upcoming appointments, builds notification pipelines (email/SMS/push), and dispatches reminders to patients and optionally to dermatologists. | |
| **Actors:** | * **Primary:** System Scheduler * **Secondary:** Patient, Dermatologist | |
| **Related use cases:** | * UC13 Appointment Booking & Management * UC17 Emergency Appointment Handling | |
| **Stakeholders:** | * Patient (avoids no-shows) * ClinicAdmin (reduces empty slots) * Dermatologist (prepares schedule) | |
| **Preconditions:** | * Appointment status = Scheduled * Patient contact information on file * Notification channels configured | |
| **Postconditions:** | * Reminder record logged * Notifications dispatched via configured channels | |
| **Flow of activities:** | **Actor** | **System** |
| 1. Trigger reminder job (e.g. at 8 AM) | 1. - 2. Query appointments where startTime = now + 24 hrs 3. For each appointment:    1. Instantiate Notification pipeline    2. send("Reminder: your appointment at …") 4. Log each sent reminder |
| **Exception conditions:** | * **Missing contact info:** skip reminder, log warning “No contact for patient X.” * **Notification send failure:** retry up to 3×, then log error and alert admin**.** | |

Figure 18: Fully developed use case description for ***Automated Appointment Reminders***

|  |  |  |
| --- | --- | --- |
| **Use case name:** | **UC19 – Dermatologist Evaluation** | |
| **Scenario:** | A dermatologist reviews a patient's skin data and provides diagnostic evaluation. | |
| **Triggering event:** | A scheduled or in-progress appointment is opened by the dermatologist. | |
| **Brief description:** | The dermatologist accesses the patient's appointment record, reviews clinical history, uploaded/AI-analyzed skin images, and inputs clinical notes and diagnosis. | |
| **Actors:** | * **Primary:** Dermatologist * **Secondary:** Patient (indirect), EHR system | |
| **Related use cases:** | * UC13 Appointment Booking & Management * UC14 AI-Powered Skin Analysis * UC20 Annotate Skin Images * UC24 Approve AI suggestions | |
| **Stakeholders:** | * Dermatologist (needs full context to evaluate) * Patient (expects accurate diagnosis and care) * Clinic (ensures quality of care) | |
| **Preconditions:** | * Appointment status = In Progress * Patient data and images are available in the system | |
| **Postconditions:** | * Evaluation notes saved * Diagnosis added to patient record * Next action triggered (e.g. annotate, prescribe, schedule follow-up) | |
| **Flow of activities:** | **Actor** | **System** |
| 1. Opens current patient session 2. – 3. Reviews and interprets data 4. Inputs diagnostic evaluation | 1. - 2. Loads patient record, skin images, history, AI report 3. - 4. - 5. Saves clinical notes & diagnosis 6. 6. Logs evaluation time and user ID |
| **Exception conditions:** | * **Data unavailable:** show “Patient data not found – contact admin.” * **Session timeout**: request re-login. * **Save failure:** notify “Evaluation not saved – try again.” | |

Figure 19: Fully developed use case description for ***Dermatologist Evaluation***

|  |  |  |
| --- | --- | --- |
| **Use case name:** | **UC20 – Annotate Skin Images** | |
| **Scenario:** | A dermatologist adds visual annotations to patient skin images for diagnostic, educational, or treatment purposes. | |
| **Triggering event:** | During or after evaluation, dermatologist clicks “Annotate” on a skin image. | |
| **Brief description:** | The system allows the dermatologist to highlight, mark, and label specific areas of skin images, storing them in the patient’s EHR for reference or further analysis. | |
| **Actors:** | * **Primary:** Dermatologist * **Secondary**: AI Analyzer (optional) | |
| **Related use cases:** | * UC14 AI-Powered Skin Analysis * UC19 Dermatologist Evaluation | |
| **Stakeholders:** | * Dermatologist (documents findings) * Patient (benefits from clarity) * Medical reviewers (for quality and second opinions) | |
| **Preconditions:** | * At least one image available in the system * User has annotation privileges | |
| **Postconditions:** | * Annotated image saved in patient record * Linked to evaluation or diagnosis | |
| **Flow of activities:** | **Actor** | **System** |
| 1. Clicks “Annotate” on image 2. Draws on/labels areas of image 3. - 4. Clicks “Save Annotations” | 1. - 2. - 3. Tracks and stores annotation shapes & metadata 4. - 5. Saves annotated version and links to patient’s EHR 6. Displays confirmation message |
| **Exception conditions:** | * **Annotation tool error**: display “Tool not responding – refresh.” * **Save failure:** “Annotation not saved – check your connection.” * **Image locked/read-only:** “Image cannot be annotated – contact admin.” | |

Figure 20: Fully developed use case description for ***Annotate Skin Images***