## Test Cases:

### SignUp Test case No.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Fundus Image Analysis/Patient/Signup Module | | | | | |
| FYP II Documentation Section 4.3 | | | | | |
| Test Case ID: | | *TA-01* | Test Date: | | *3-4-2025* |
| Test case Version: | | *V1.0* | Use Case Reference(s): | | *3.4.1 Sign-Up* |
| Revision History: | | *None* | | | |
| Objective | | *To test the Sign-Up module for a patient, ensuring the system validates inputs, creates a new account, sends an email verification link, and handles invalid/duplicate inputs as per use case 3.4.1.* | | | |
| Product/Ver/Module: | | *FundusImageAnalysis/Patient/Sign-Up* | | | |
| Environment: | | *PC/Browser (e.g., Chrome, Firefox)* | | | |
| Assumptions: | | *The patient has not already signed up with the same email address.* | | | |
| Pre-Requisite: | | *The patient must have access to the DermaAI Sign-Up page via the web application.* | | | |
| Step No. | Execution description | | | Procedure result | |
|  | *The patient opens the Sign-Up page and fills in the required fields with valid data:*  *- Name: Alphabetic characters only (e.g., John Doe)*  *- Email: Valid format with '@' and domain (e.g., john.doe@example.com)*  *- Password: At least 6 characters (e.g., secure123)*  *- Role: Selects "Patient"* | | | *The system validates the inputs, creates a new user account, sends an email verification link to the provided email address, and displays a success message: "Registration successful, please verify your email."*  *.* | |
|  | *The patient submits the form with invalid data:*  *- Name: Contains numbers or special characters (e.g., John123@)*  *- Email: Missing '@' or invalid format (e.g., johndoe)*  *- Password: Less than 6 characters (e.g., pass)* | | | *The system displays error messages for each invalid field: "Invalid name: Use alphabetic characters only," "Invalid email: Must contain '@' and a valid domain," "Password must be at least 6 characters long." The account is not created.* | |
|  | *The patient submits the form with an email already registered in the system (e.g., john.doe@example.com, previously used).* | | | *The system displays an error message: "Email already in use. Please use a different email or log in." The account is not created.* | |
| Comments: The Sign-Up module only accepts valid input formats as specified in use case 3.4.1. Duplicate emails are rejected, ensuring unique user accounts. The system successfully enforces email verification, aligning with the post-condition of the use case. | | | | | |
| *Passed* *Failed* *Not Executed* | | | | | |

### Login Test Case No.2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Fundus Image Analysis/Patient/Login Module | | | | | |
| FYP II Documentation Section 4.3 | | | | | |
| Test Case ID: | | *TA-02* | Test Date: | | *7-5-2025* |
| Test case Version: | | *V1.0* | Use Case Reference(s): | | *3.4.2 Login* |
| Revision History: | | *None* | | | |
| Objective | | *To test the Login module for all user types (Patient/User, Doctor, Admin), ensuring the system correctly validates credentials, handles errors, and redirects users to their respective dashboards as per Use Case 3.4.2.* | | | |
| Product/Ver/Module: | | *Fundus Image Analysis / Multi-Role / Login* | | | |
| Environment: | | *PC/Browser (e.g., Chrome, Firefox)* | | | |
| Assumptions: | | *Users have already registered and their emails are verified.* | | | |
| Pre-Requisite: | | *The user must access the DermaAI Login page and have valid credentials..* | | | |
| Step No. | Execution description | | | Procedure result | |
|  | *The user navigates to the Login page.* | | | *The system displays a login form with email and password fields..* | |
|  | *The user enters valid email and password and submit* | | | *The system authenticates the user and redirects them to their respective dashboard (Patient, Doctor, or Admin).* | |
|  | |  | | --- | |  |  |  | | --- | | *The user enters invalid credentials*  *(wrong password or email).* | | | | *The system displays an error message: "Invalid email or password. Please try again."* | |
|  | *The user enters valid credentials but email is not verified.* | | | *The system displays: "Please verify your email before logging in."* | |
|  | *The user clicks "Forgot Password" link.* | | | *The system redirects to the password recovery page/process.* | |
|  | *The login request is made over an insecure (non-HTTPS) connection.* | | | *The system blocks the request or redirects to HTTPS, ensuring secure login.* | |
|  | |  | | --- | |  |  |  | | --- | | *The user attempts login with a*  *deactivated or deleted account.* | | | | *The system displays: "Account is inactive. Please contact support."* | |
| Comments: This test case covers the normal and alternate flows defined in the Login use case (3.4.2). It ensures role-based redirection, secure authentication, and robust error handling. | | | | | |

.

.

### Create Profile Test case No.3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Fundus Image Analysis/Patient/ Create Profile Module | | | | | |
| FYP II Documentation Section 4.3 | | | | | |
| Test Case ID: | | |  | | --- | |  |  |  | | --- | | ***TA-03*** | | Test Date: | | *07-05-2025* |
| Test case Version: | | *V1.0* | Use Case Reference(s): | | *3.4.3 – Create Profile* |
| Revision History: | | *None* | | | |
| Objective | | *To test the Create Profile module for both User (Patient/Doctor) and Admin roles, ensuring accurate data input, proper validation, and successful database entry as per Use Case 3.4.3.* | | | |
| Product/Ver/Module: | | *Fundus Image Analysis / Multi-Role / Create Profile* | | | |
| Environment: | | *PC/Browser (e.g., Chrome, Firefox)* | | | |
| Assumptions: | | *User or Admin is already logged in and authenticated.* | | | |
| Pre-Requisite: | | *Access to the Create Profile section is available after login.* | | | |
| Step No. | Execution description | | | Procedure result | |
|  | *The logged-in user navigates to the "Create Profile" page.* | | | *The system displays a form with required fields (e.g., name, contact, history).* | |
|  | *The user fills the form with valid data: name, contact number, address, and history/qualification.* | | | *The system validates all inputs and highlights no errors.* | |
|  | |  | | --- | |  |  |  | | --- | | *The user submits the completed form.* | | | | *The system saves the profile to the database and shows: "Profile created successfully.* | |
|  | *The user submits the form with invalid data (e.g., name: “Ali123!”, phone: “abc123”).* | | | *The system shows validation errors: "Name must only contain letters", "Phone must be numeric".* | |
|  | |  | | --- | |  |  |  | | --- | | *Admin fills profile form with professional details (e.g., name, email, designation).* | | | | *Admin profile data is validated and saved successfully.* | |
|  | *The user skips required fields and submits..* | | | |  | | --- | |  |  |  | | --- | | *The system blocks submission*  *and highlights missing required fields.* | | |
|  | |  | | --- | |  |  |  |  |  | | --- | --- | --- | | |  | | --- | |  |  |  | | --- | | *The user attempts form submission*  *without being logged in.* | | | | | *The system redirects to the login page or shows "Access Denied".* | |
| Comments: This test case checks both the normal and error-handling paths for profile creation. It ensures proper validation for multiple user types and that only authenticated users can access the feature. | | | | | |

### Edit Profile Test case No.4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Fundus Image Analysis/Patient/ Edit Profile Module | | | | | |
| FYP II Documentation Section 4.3 | | | | | |
| Test Case ID: | | |  | | --- | |  |  |  | | --- | | ***TA-04*** | | Test Date: | | *07-05-2025* |
| Test case Version: | | *V1.0* | Use Case Reference(s): | | |  | | --- | |  |  |  | | --- | | *3.4.4 Edit Profile* | |
| Revision History: | | *None* | | | |
| Objective | | *To verify that users (Patients, Doctors, Admins) can successfully edit their profiles, ensuring data validation, proper updates in the database, and appropriate error handling for invalid inputs.* | | | |
| Product/Ver/Module: | | *Fundus Image Analysis / User-Doctor-Admin / Edit Profile* | | | |
| Environment: | | *PC / Browser (Chrome, Firefox)* | | | |
| Assumptions: | | *The user is logged in and has an existing profile created.* | | | |
| Pre-Requisite: | | *User must have access to the Edit Profile page with previously saved data preloaded.* | | | |
| Step No. | Execution description | | | Procedure result | |
|  | *The user navigates to the Edit Profile page.* | | | |  | | --- | |  |  |  | | --- | | *System fetches and displays existing*  *profile information.* | | |
|  | *The user edits valid fields (e.g., name, contact, medical history).* | | | *System accepts the valid input and enables the Save/Update button.* | |
|  | |  | | --- | |  |  |  | | --- | | *The user clicks on "Save" after*  *updating details.* | | | | *System updates the profile in the database and shows "Profile updated successfully."* | |
|  | *The user leaves mandatory fields empty or enters invalid data (e.g., phone number in name field).* | | | *System displays appropriate error messages like "Name must only contain letters," "Email format is invalid."* | |
|  | |  | | --- | |  |  |  | | --- | | *The user submits a form without*  *making any changes.* | | | | *System notifies: "No changes detected." Profile remains unchanged.* | |
| Comments: The Edit Profile feature ensures only valid data is accepted. Input validation is strict, maintaining the integrity of user records. Error messages are informative, and the user experience is smooth. | | | | | |

### Email Verification Test case No.5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Fundus Image Analysis/Patient/ Email Verification Module | | | | | |
| FYP II Documentation Section 4.3 | | | | | |
| Test Case ID: | | |  | | --- | |  |  |  | | --- | | ***TA-05*** | | Test Date: | | *07-05-2025* |
| Test case Version: | | *V1.0* | Use Case Reference(s): | | |  | | --- | |  |  |  | | --- | | *3.4.5 Email Verification* | |
| Revision History: | | *None* | | | |
| Objective | | *To ensure that after sign-up, a verification email is sent to the user, and that the verification process activates the user account successfully.* | | | |
| Product/Ver/Module: | | *Fundus Image Analysis / Email Verification* | | | |
| Environment: | | *PC / Browser (Chrome), Gmail/Yahoo/Outlook Inbox, Internet* | | | |
| Assumptions: | | *Email service (SMTP/NodeMailer or other) is integrated and functional.*  *The user provides a valid email address during registration.* | | | |
| Pre-Requisite: | | *A user account is registered but unverified.*  *The system can send emails to the registered email address.* | | | |
| Step No. | Execution description | | | Procedure result | |
|  | *The user completes and submits the registration form.* | | | |  | | --- | |  |  |  | | --- | | *System registers the user and sends a*  *verification email.* | | |
|  | *The user checks their inbox and clicks the verification link.* | | | *System verifies the token and marks the email as verified.* | |
|  | |  | | --- | |  |  |  | | --- | | *The user attempts to log in before*  *verifying their email.* | | | | *System restricts access and prompts: “Please verify your email.”* | |
|  | *The user clicks an expired or invalid verification link.* | | | *System displays: “Verification link expired. Request a new one.”* | |
|  | |  | | --- | |  |  |  |  |  | | --- | --- | --- | | |  | | --- | |  |  |  | | --- | | *The user clicks on*  *"Resend verification email."* | | | | | *System sends a new email and confirms the resend.* | |
|  | *The email service is down temporarily.* | | | |  | | --- | |  |  |  | | --- | | *System shows: “Email service unavailable. Please try again later.”* | | |
| Comments: Email verification ensures account authenticity and improves security. Each alternate or exception flow is handled with clear feedback to the user. | | | | | |

### Forgot Password Test case No.6

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Fundus Image Analysis/Patient/Signup Module | | | | | |
| FYP II Documentation Section 4.3 | | | | | |
| Test Case ID: | | |  | | --- | |  |  |  | | --- | | ***TA-06*** | | Test Date: | | *07-05-2025* |
| Test case Version: | | *V1.0* | Use Case Reference(s): | | |  | | --- | |  |  |  | | --- | | *3.4.6 Forgot Password* | |
| Revision History: | | *None* | | | |
| Objective | | *To validate the Forgot Password functionality, ensuring users can securely reset their password through email verification.* | | | |
| Product/Ver/Module: | | *Fundus Image Analysis / Forgot Password Module* | | | |
| Environment: | | *Web App on Chrome/Edge | Email client (Gmail/Yahoo) | Active Internet* | | | |
| Assumptions: | | *The user has already registered with a valid and accessible email. Email sending services (e.g., NodeMailer, SMTP) are properly configured.* | | | |
| Pre-Requisite: | | *The user has forgotten their password and initiates a password reset process via the login page.* | | | |
| Step No. | Execution description | | | Procedure result | |
|  | |  | | --- | |  |  |  | | --- | | *User clicks “Forgot Password” on*  *login page.* | | | | |  | | --- | |  |  |  | | --- | | *System shows a form asking for email.* | | |
|  | *User enters registered email.* | | | |  | | --- | |  |  |  | | --- | | *System validates and sends password*  *reset link.* | | |
|  | |  | | --- | |  |  |  | | --- | | *User opens the email and clicks the*  *reset link or enters OTP.* | | | | *System verifies the token and redirects to password reset form.* | |
|  | *User enters and confirms a new password.* | | | |  | | --- | |  |  |  | | --- | | *System validates the match.* | | |
|  | |  | | --- | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  | | --- | |  |  |  |  |  | | --- | --- | --- | | |  | | --- | |  |  |  | | --- | | *User submits the new password.* | | | | | | |  | | --- | |  |  |  | | --- | | *System updates password and shows*  *success message.* | | |
|  | |  | | --- | |  |  |  | | --- | | *User logs in with new password.* | | | | |  | | --- | |  |  |  |  |  | | --- | --- | --- | | |  | | --- | |  |  |  | | --- | | *System grants access.* | | | |
| Comments: All main success, alternative, and exception flows are covered. System behavior is clear and user-friendly even in failure cases. | | | | | |

### Logout Test case No.7

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Fundus Image Analysis/Patient/Signup Module | | | | | |
| FYP II Documentation Section 4.3 | | | | | |
| Test Case ID: | | |  | | --- | |  |  |  | | --- | | ***TA-07*** | | Test Date: | | *07-05-2025* |
| Test case Version: | | *V1.0* | Use Case Reference(s): | | |  | | --- | |  |  |  | | --- | | *3.4.7 Logout* | |
| Revision History: | | *None* | | | |
| Objective | | *To verify that users (Admin, Doctor, Patient/User) can securely log out and that their session ends appropriately.* | | | |
| Product/Ver/Module: | | *Fundus Image Analysis / Logout Module* | | | |
| Environment: | | *Web App – Browser: Chrome, Edge | Session Storage/Token-based Auth* | | | |
| Assumptions: | | *User is already logged in.*  *Authentication tokens or sessions are active.* | | | |
| Pre-Requisite: | | *A logged-in session must exist before logout is initiated.* | | | |
| Step No. | Execution description | | | Procedure result | |
|  | |  | | --- | |  |  |  | | --- | | *User clicks on the “Logout” button/link.* | | | | *System initiates session termination.* | |
|  | *System clears session/token from storage.* | | | |  | | --- | |  |  |  | | --- | | *User session is destroyed.* | | |
|  | |  | | --- | |  |  |  |  |  | | --- | --- | --- | | |  | | --- | |  |  |  | | --- | | *System redirects user to login or*  *home page.* | | | | | |  | | --- | |  |  |  | | --- | | *System shows confirmation message: “You have been logged out.”* | | |
|  | *User tries to navigate to dashboard or restricted page.* | | | |  | | --- | |  |  |  |  |  | | --- | --- | --- | | |  |  | | --- | --- | |  | System redirects back to loggin due to invalid session. | | | |
| Comments: Logout should be secure, remove sensitive tokens, and prevent re-entry without login. Even if session expired, system should gracefully handle it. | | | | | |

### Search Doctor Test case No.8

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Fundus Image Analysis/Patient/ Search Doctor Module | | | | | |
| FYP II Documentation Section 4.3 | | | | | |
| Test Case ID: | | |  | | --- | |  |  |  | | --- | | ***TA-08*** | | Test Date: | | *07-05-2025* |
| Test case Version: | | *V1.0* | Use Case Reference(s): | | |  | | --- | |  |   *3.4.8 Search Doctor* |
| Revision History: | | *None* | | | |
| Objective | | *To verify that patients can successfully search for doctors using filters like specialization, location, and availability, and receive appropriate results or feedback.* | | | |
| Product/Ver/Module: | | *DoctorSearch / Web Portal* | | | |
| Environment: | | *Web App – Browser: Chrome, Edge | Session Storage/Token-based Auth* | | | |
| Assumptions: | | *Patient is logged in.*  *Doctor profiles exist in the system database.* | | | |
| Pre-Requisite: | | *Patient has access to the "Search Doctor" page.* | | | |
| Step No. | Execution description | | | Procedure result | |
|  | |  | | --- | |  |  |  | | --- | | *Patient navigates to "Search Doctor".* | | | | *Search form with filters is displayed.* | |
|  | *Patient selects specialization (e.g., Cardiologist) and enters location* | | | *Fields are accepted without error.* | |
|  | |  | | --- | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  | | --- | |  |  |  |  |  | | --- | --- | --- | | |  | | --- | |  |  |  | | --- | | *Patient clicks "Search" button.* |   *.* | | | | | |  | | --- | |  |  |  | | --- | | *Form data is sent to the backend.* | | |
|  | *System queries doctor database based on filters.* | | | |  | | --- | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  |  | | --- | --- | --- | --- | |  | |  | | --- | |  |  |  | | --- | | Matching doctor list is retrieved. | | | | |
|  | *System displays search results in a list or card format.* | | | *Doctors are shown with name, specialty, and availability* | |
| Comments: The test case and matrix for "Search Doctor" effectively validate all user scenarios and system responses. With a 100% pass rate, it confirms accurate filtering and robust handling of edge cases. | | | | | |

### Upload Eye Images Test case No.9

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Fundus Image Analysis/Patient/ Upload Eye Images Module | | | | | |
| FYP II Documentation Section 4.3 | | | | | |
| Test Case ID: | | |  | | --- | |  |  |  | | --- | | ***TA-09*** | | Test Date: | | *07-05-2025* |
| Test case Version: | | *V1.0* | Use Case Reference(s): | | |  | | --- | |  |  |  | | --- | | *3.4.8 Upload Eye Images* | |
| Revision History: | | *None* | | | |
| Objective | | *To verify that a patient can successfully upload valid eye images and the system handles file validation, errors, and storage properly.* | | | |
| Product/Ver/Module: | | *Fundus Image Analysis / Image Upload* | | | |
| Environment: | | *Web App – Chrome/Edge, laravel Backend with php, Secure Cloud Storage (if applicable)* | | | |
| Assumptions: | | *Patient is logged in.*  *Upload component is functional.*  *Server is connected to secure storage* | | | |
| Pre-Requisite: | | *Upload page is accessible.*  *Patient has valid eye image(s) in supported formats and sizes.* | | | |
| Step No. | Execution description | | | Procedure result | |
|  | *Patient navigates to the "Upload Eye Images" section.* | | | |  | | --- | |  |  |  | | --- | | *Upload interface is displayed.* | | |
|  | |  | | --- | |  |  |  | | --- | | *Patient selects image(s) in .jpg/.png format.* | | | | |  | | --- | |  |  |  |  |  | | --- | --- | --- | | |  | | --- | |  |  |  | | --- | | *Image(s) appear in preview list.* | | | |
|  | |  | | --- | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  | | --- | |  |  |  |  |  | | --- | --- | --- | | |  | | --- | |  |  |  | | --- | | *Patient clicks the "Upload" button.* | | | | | | |  | | --- | |  |  |  | | --- | | Form is submitted. | | |
|  | *System validates image size and format.* | | | |  | | --- | |  | | If valid, proceeds. | | |  |  |  | | --- | --- | | |  | | --- | |  | | | |
|  | |  | | --- | |  |  |  | | --- | | *System stores image securely.* | | | | *Image is saved in database or cloud.* | |
|  | *System confirms successful upload.* | | | |  | | --- | |  |  |  | | --- | | *Message: “Image uploaded successfully.”* | | |
|  | |  | | --- | |  |  |  | | --- | | *Image is linked to patient profile.* | | | | |  | | --- | |  |  |  | | --- | | *Patient can see uploaded image in history.* | | |
| Comments: Logout should be secure, remove sensitive tokens, and prevent re-entry without login. Even if session expired, system should gracefully handle it. | | | | | |

### Book Appointment Test case No.10

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Fundus Image Analysis/Patient/ Book Appointment Module | | | | | |
| FYP II Documentation Section 4.3 | | | | | |
| Test Case ID: | | |  | | --- | |  |  |  | | --- | | ***TA-09*** | | Test Date: | | *07-05-2025* |
| Test case Version: | | *V1.0* | Use Case Reference(s): | | |  | | --- | |  |  |  |  | | --- | --- | | |  | | --- | |  |   *3.4.10 Book Appointment* | |
| Revision History: | | *None* | | | |
| Objective | | *To verify that a logged-in patient can successfully book an appointment with a selected doctor, view available slots, and handle cases such as no slots or booking failure.* | | | |
| Product/Ver/Module: | | *AppointmentBooking / Web Portal* | | | |
| Environment: | | *Web App – Chrome/Edge, laravel Backend with php, Secure Cloud Storage (if applicable)* | | | |
| Assumptions: | | *Patient is logged in.*  *Doctor profiles with available slots are available in the database.* | | | |
| Pre-Requisite: | | *Patient has access to the "Book Appointment" interface via selected doctor.* | | | |
| Step No. | Execution description | | | Procedure result | |
|  | *Patient selects a doctor and navigates to "Book Appointment"..* | | | |  | | --- | |  |  |  | | --- | | *Available slots for that doctor are shown.* | | |
|  | *Patient selects a time slot and submits the form.* | | | |  | | --- | |  |  |  |  |  | | --- | --- | --- | | |  | | --- | |  |  |  | | --- | | *Selected slot is captured and validated..* | | | |
|  | |  | | --- | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  | | --- | |  |  |  |  |  | | --- | --- | --- | | |  | | --- | |  |  |  | | --- | | *Backend checks for slot availability and confirms.* | | | | | | *Appointment is booked and stored in DB.* | |
|  | |  | | --- | |  |  |  | | --- | | *System displays confirmation message.* | | | | |  | | --- | |  | | |  | | --- | |  |  |  | | --- | | Patient sees "Appointment booked successfully." | | | |  |  |  | | --- | --- | | |  | | --- | |  | | | |
| Comments: The use case clearly defines the process for booking appointments, covering both normal and exceptional scenarios. It ensures validation of doctor availability and provides user-friendly feedback. | | | | | |