Epics, User Stories, UX/UI Architecture, Information Architecture, Analytical Reporting

Team Eagles

# Table of Contents

[Table of Contents 2](#_heading=h.30j0zll)

[1.](#_heading=h.1fob9te) OVERVIEW 2

[2.](#_heading=h.id1pnt52y7lw) Description of the problem in your words 3

[3.](#_heading=h.3znysh7) Key Features 3

[4.](#_heading=h.tyjcwt) System Context Diagram 3

[5.](#_heading=h.3dy6vkm) Requirements: Epics and User Stories 4

[6.](#_heading=h.kb6c1ts2leke) Quality Attributes 5

[6.1 user stories to quality attributes map 5](#_heading=h.3whwml4)

[6.2 User Stories grouping based on Quality Attributes 6](#_heading=h.qsh70q)

[7.](#_heading=h.3as4poj) User experience and user interface 6

[7.1 Value proposition canvas 6](#_heading=h.1pxezwc)

[7.2 Create wireframes for important user interactions 7](#_heading=h.49x2ik5)

[7.3 Build UX/UI architecture 7](#_heading=h.2p2csry)

[8.](#_heading=h.eg6nvoxnhbl2) Information Architecture 7

[8.1 Data Entities to User Stories map 8](#_heading=h.bozfr3ff9f5g)

[8.2 Data Architecture Diagrams and description 8](#_heading=h.icf4pvjvypn1)

[8.3 Analytical Reporting and Architecture 8](#_heading=h.rxldr99b57ah)

[9.](#_heading=h.bf6upr52rrpm) Risks 9

[10.](#_heading=h.8q0q89xps5si) System level Architectural choices 9

[11.](#_heading=h.32hioqz) Applicable Standards 9

[12.](#_heading=h.1y810tw) Definitions, Acronyms, Initialisms, and Abbreviations (DAIAs) 9

[13.](#_heading=h.4i7ojhp) References 9

## OVERVIEW

This document outlines the key foundational facets of the architecture that are required for the solution. These facets, or “views” of the architecture are constructed to support the requirements as defined in the problem statement and user stories. This document is focused primarily on capturing details of the external view of the system as the user will experience and use it:

* **System Context**
* **Requirements**
* **Quality Attributes**
* **User Experience and User Interface Architecture**
* **Information Architecture**

The requirements and architecture defined in this document will inform the internal system architecture views and Execution plan to be defined in Deliverable 2, which include Software Architecture, Security Architecture, Deployment Infrastructure and Execution Plan.

## Description of the problem in your words

### iCare is a non-profit organization aiming to provide affordable prescription glasses to people with Uncorrected Refractive Error (URE) in India, where 100M people and 48M children cannot afford eyeglasses. The organization needs a comprehensive technology platform to manage its operations, track impact, ensure sustainability, and maintain compliance while serving a large population across urban and rural areas. The platform must handle various aspects from service delivery to fundraising, while ensuring data privacy and measuring social impact.

## Key Features

**Minimum Viable Product**

### Vision Assessment & Screening

Our platform offers comprehensive self-administered vision tests powered by AI analysis technology. Users can quickly assess their visual acuity, color perception, and astigmatism through intuitive tests designed for all literacy levels. The system provides immediate results with clear interpretation and recommends the next steps. Understanding connectivity challenges in many regions, we've implemented a robust offline mode that stores test data for later synchronization. All interfaces and instructions support multiple Indian languages with supplementary audio guidance to ensure accessibility for all users.

* Appointment Management

The appointment system streamlines access to eye care through intelligent provider matching. Users can search for optometrists based on location and availability. Our scheduling system integrates seamlessly with calendar applications and sends timely reminders via SMS to reduce no-shows. Before appointments, users receive customized preparation instructions, and afterward, they can provide feedback to improve service quality continuously. This end-to-end approach minimizes barriers to professional care.

* Prescription & Eyewear Management

After diagnosis, users benefit from secure digital prescription storage and tracking. The platform includes a basic eyewear catalog with filtering options based on style, price range, and availability. Order placement is straightforward, with standard delivery tracking to ensure transparency. For eligible users, the system provides subsidy verification and streamlined application processes. Once eyewear is received, users access comprehensive usage instructions and care guidelines to maximize the lifespan and effectiveness of their glasses.

### User Profile & Records

The platform prioritizes user-friendly registration with multi-language support. Each profile securely stores medical history and maintains a comprehensive record of prescriptions and vision test results.

Future Enhancements

* Advanced Diagnostics

Future versions will incorporate additional specialized vision tests and AI-driven progression prediction to identify potential issues before they become serious. We plan to integrate with professional diagnostic equipment in vision centers and enable remote monitoring capabilities for providers. Personalized vision improvement exercises will supplement corrective eyewear for comprehensive vision care.

* Enhanced Supply Chain

We aim to develop end-to-end order tracking with real-time updates throughout the manufacturing and delivery process. Advanced inventory management across locations will optimize resource allocation. Users will benefit from virtual try-on technology for eyewear selection and options for sustainable materials. The system will eventually support automated reordering based on wear patterns and prescription changes.

* Family Vision Management

We will implement comprehensive family management features that allow parents to oversee children's eye care and elderly family members' vision health. These features will include family vision dashboards, consolidated appointment management, shared access controls, hereditary condition tracking, and school vision program integration.

* Community & Education Platform

Building on our foundation, we'll develop peer support communities organized by region and condition. A comprehensive educational content library will offer personalized recommendations based on user profiles. Gamified compliance tracking will encourage adherence to wearing schedules, while dedicated tools for school screening programs will expand our reach.

System Boundaries

* Medical Treatment Limitations

Our system focuses exclusively on refractive error correction and does not provide direct medical interventions or procedures. The platform will not diagnose non-refractive eye conditions, offer emergency eye care services, manage pharmaceutical prescriptions, or support surgical planning. Users requiring these services will be appropriately referred to medical professionals.

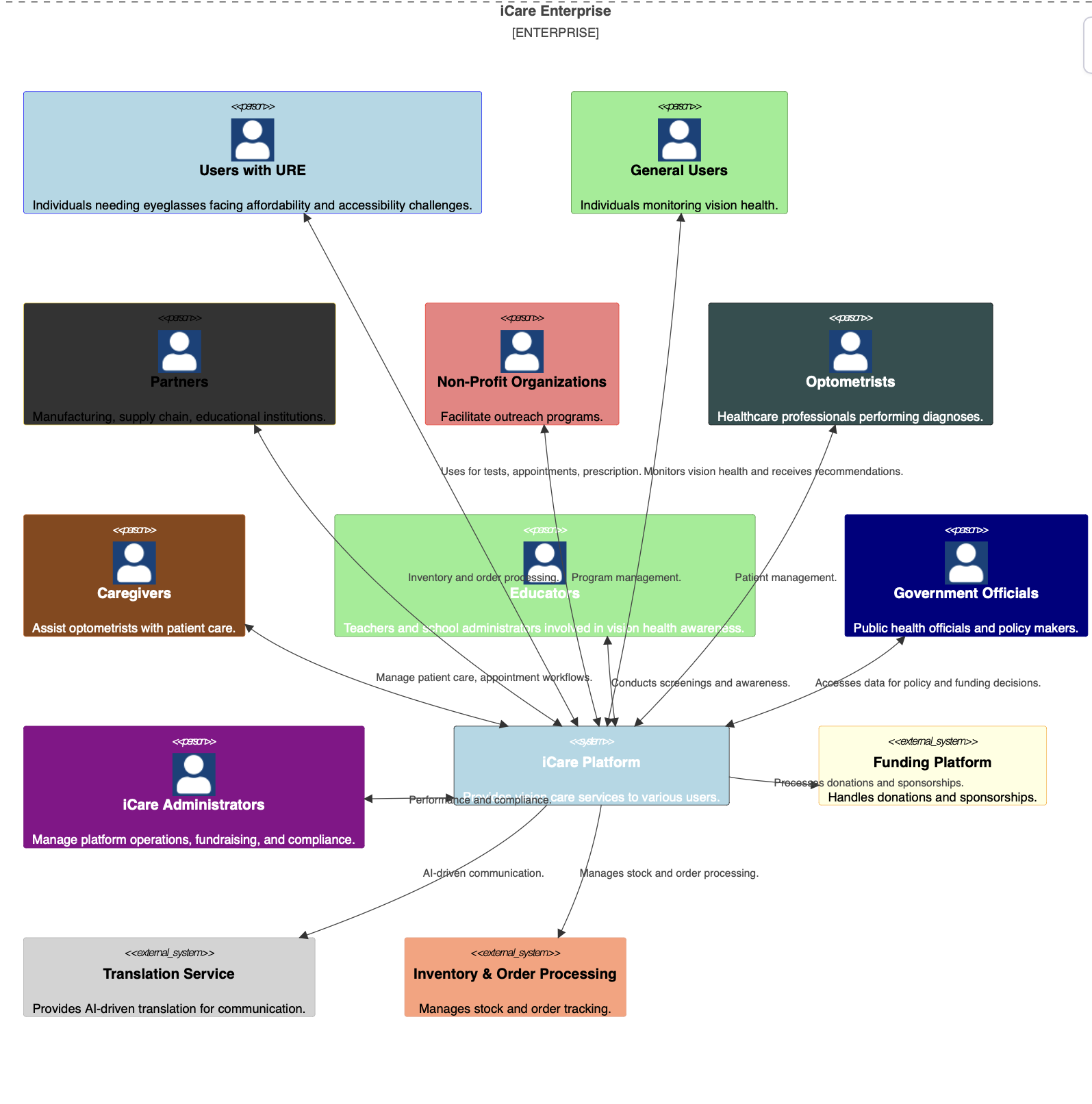
* Hardware Scope Restrictions

The platform leverages existing consumer devices and does not involve custom diagnostic hardware development, eyewear manufacturing, specialized medical equipment, custom mobile devices, or physical infrastructure for vision centers. We maximize accessibility by working with widely available technology rather than requiring proprietary hardware.

* General Healthcare Exclusions

While comprehensive within vision care, our system does not manage non-vision health conditions, integrate with general electronic health records, process insurance claims, handle billing for medical services, or manage staffing for healthcare facilities. This focused approach ensures excellence in our core mission of addressing uncorrected refractive error.

## System Context Diagram



## Requirements: Epics and User Stories

This section outlines the **Epics** and **User Stories** that define the functional requirements of the iCare platform. Each epic represents a major capability of the system, and user stories within each epic break down specific user needs.

**Epics / User Stories**

**Epic 1: Vision Screening & Diagnosis**Description: Provide users with tools for online vision testing and facilitate access to in-person screenings through camps and optometrists.

User Stories:

User Story 1.1: As a user with URE, I want to take an online vision test so that I can determine if I need prescription glasses.

User Story 1.2: As a general user, I want to check my vision for preventive measures so that I can take action if necessary.

User Story 1.3: As a rural user, I want mobile eye checkup camps near my location so that I don’t have to travel long distances for an exam.

User Story 1.4: As a parent, I want to check my child’s vision regularly through the platform so that I can take preventive action if needed.

User Story 1.5: As a visually impaired user, I want audio-based guidance in my local language so that I can complete vision tests without reading text.

User Story 1.6: As a user, I want to be notified about upcoming free eye check-up camps so that I can plan to attend.

**Epic 2: Appointment Scheduling & Optometrist Interaction**Description: Enable users to book appointments, receive reminders, and facilitate optometrist-patient communication.

User Stories:

User Story 2.1: As a user with URE, I want to book an appointment with an optometrist through the platform so that I can receive a professional eye checkup.

User Story 2.2: As a patient, I want reminders for my scheduled appointment so that I don’t miss my consultation.

User Story 2.3: As an optometrist, I want to manage patient appointments efficiently so that I can reduce wait times.

User Story 2.4: As a caregiver, I want to coordinate follow-up appointments so that patients receive continuous care.

User Story 2.5: As a healthcare provider, I want AI-driven live translation to facilitate communication so that language barriers do not hinder treatment.

**Epic 3: Prescription Management & Eyeglasses Ordering**

Description: Allow users to manage prescriptions digitally and track eyeglasses orders.

User Stories:

User Story 3.1: As a diagnosed user, I want my prescription to be saved digitally so that I can easily access it for future use.

User Story 3.2: As a low-income patient, I want to check if I qualify for subsidized or donated glasses so that I can afford corrective lenses.

User Story 3.3: As a patient, I want to track my eyeglasses order status so that I know when they will arrive.

User Story 3.4: As an optometrist, I want to submit digital prescriptions for suppliers to process quickly.

**Epic 4: Multilingual & Inclusive Access**

Description: Ensure that users from diverse backgrounds can access vision care services in their preferred language and format.

User Stories:

User Story 4.1: As a non-English-speaking user, I want the platform to support my local language so that I can navigate it easily.

User Story 4.2: As a user who speaks a regional dialect, I want the online vision test to provide instructions in my preferred language so that I can complete it without confusion.

User Story 4.3: As a rural user, I want appointment instructions and vision care information in my native language so that I can understand them.

User Story 4.4: As a visually impaired user, I want audio-based guidance in my local language so that I can complete vision tests without reading text.

User Story 4.5: As an optometrist, I want AI-driven live translation to communicate effectively with non-English-speaking patients.

**Epic 5: Vision Health Tracking & Reports**

Description: Allow users to track their vision health over time and receive relevant reports.

User Stories:

User Story 5.1: As a user, I want to track my vision test history so that I can see any changes over time.

User Story 5.2: As an individual with a family history of vision problems, I want insights on my risk factors so that I can take preventive steps early.

User Story 5.3: As a parent, I want to check my child’s vision periodically through the platform so that I can detect issues early.

User Story 5.4: As iCare, I want to monitor the number of vision tests completed so that I can measure the platform's reach and impact.

**Epic 6: Health & Lifestyle Recommendations**Description: Provide users with guidance on maintaining good vision health.

User Stories:

User Story 6.1: As a general user, I want to receive eye health tips and exercises so that I can maintain good vision.

User Story 6.2: As a digital device user, I want recommendations on reducing eye strain so that I can prevent vision deterioration.

**Epic 7: iCare Business Analytics & Reporting**Description: Provide iCare management with insights into platform performance, user engagement, and financial sustainability.

User Stories:

User Story 7.1: As iCare, I want to track registered optometrists and caregivers so that I can ensure adequate service coverage.

User Story 7.2: As iCare, I want to generate reports on appointments, screenings, and prescriptions so that I can assess platform performance.

User Story 7.3: As iCare, I want to track funding sources and donation trends so that I can ensure the financial sustainability of the platform.

User Story 7.4: As iCare, I want to generate impact reports for donors and sponsors so that I can attract more funding.

**Epic 8: Supply Chain & Order Management**Description: Optimize the supply chain and order management processes for manufacturers, suppliers, and logistics providers.

User Stories:

User Story 8.1: As a manufacturing partner, I want to receive automated purchase orders based on AI-driven demand forecasting so that I can optimize production schedules.

User Story 8.2: As a supplier, I want to update inventory levels in real-time so that the platform accurately reflects available stock.

User Story 8.3: As a logistics provider, I want to integrate my tracking system with the platform so that all stakeholders can monitor shipment status.

**Epic 9: Compliance & Quality Control**Description: Ensure quality control in manufacturing and compliance with industry standards.

User Stories:

User Story 9.1: As a lens manufacturer, I want to record and track quality metrics for each batch so that I can maintain high production standards.

User Story 9.2: As a frame supplier, I want to upload certification documents to comply with regulatory requirements.

User Story 9.3: As a distributor, I want to maintain a chain of custody record so that I can ensure product traceability.

**Epic 10: Government & Policy Integration**Description: Provide data and analytics to policymakers to support public health initiatives.

User Stories:

User Story 10.1: As a government official, I want to access real-time data on URE prevalence so that I can make informed healthcare policy decisions.

User Story 10.2: As a government health officer, I want to track activity in different regions so that I can support operations with efficient funding through direct initiatives or NGOs.

User Story 10.3: As a policymaker, I want to analyze vision health trends in schools and communities so that I can develop targeted intervention programs.

**Epic 11: Non-Profit & Community Engagement**Description: Enable non-profits to track impact, manage community programs, and ensure compliance.

User Stories:

User Story 11.1: As an NPO manager, I want to track the number of beneficiaries served so that I can measure program impact.

User Story 11.2: As a field coordinator, I want to map underserved areas so that we can plan mobile vision camps effectively.

User Story 11.3: As an NPO fundraiser, I want to generate detailed program metrics so that we can support funding applications.

User Story 11.4: As a program manager, I want to ensure data privacy compliance so that we can protect beneficiary information.

## Quality Attributes

Quality attributes define the non-functional requirements of the **iCare Vision Platform**, ensuring that the system meets key performance, usability, and security standards. Below are the quality attributes essential to the success of the platform:

## 6.1 user stories to quality attributes map

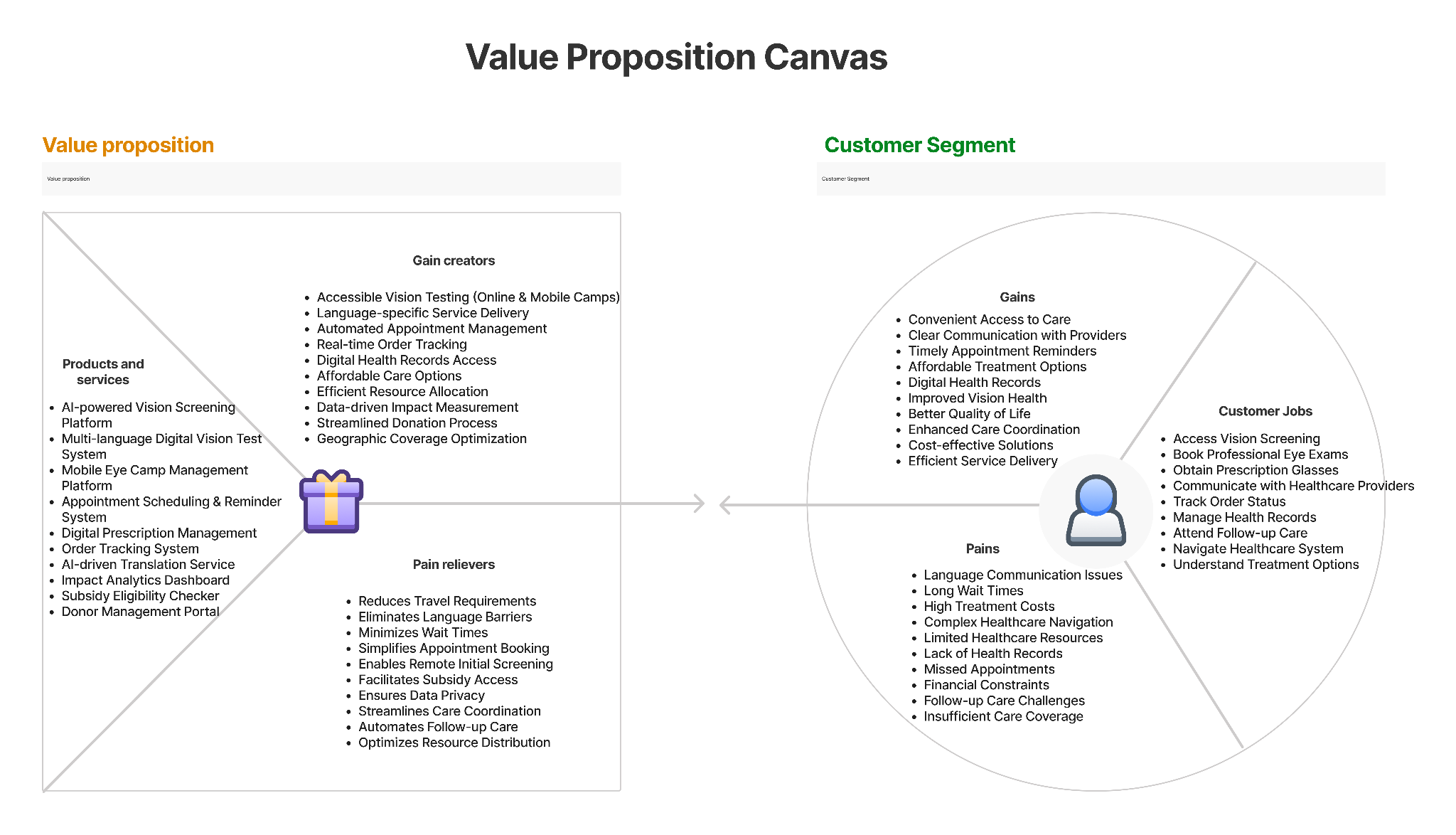
|  |  |  |
| --- | --- | --- |
| **User Stories** | **Quality attributes mapping** | **Justification for the mapping** |
| **User Story 1.1:** As a user with URE, I want to take an online vision test so that I can determine if I need prescription glasses. | Availability, Performance, Usability | The online test must be available anytime and should perform efficiently to provide accurate results. The user interface should be simple and intuitive for ease of use. |
| **User Story 1.2:** As a general user, I want to check my vision for preventive measures so that I can take action if necessary. | Availability, Security, Maintainability | The system should be available at all times, secure for storing user data, and easy to maintain for future improvements. |
| **User Story 1.3:** As a rural user, I want mobile eye checkup camps near my location so that I don’t have to travel long distances for an exam. | Scalability, Flexibility | The platform should scale to support different regions and allow integration with location-based services. |
| User Story 1.4: As a parent, I want to check my child’s vision regularly through the platform so that I can take preventive action if needed. | |  | | --- | | Security, Usability |  |  | | --- | |  | | The system should store family health records securely and provide a user-friendly interface for parents. |
| User Story 2.1: As a user with URE, I want to book an appointment with an optometrist through the platform so that I can receive a professional eye checkup. | Availability, Security | The system should be available for seamless booking, and patient data should be securely stored. |
| **User Story 2.2:** As a patient, I want reminders for my scheduled appointment so that I don’t miss my consultation. | Availability, Performance | The reminder system should be reliable and timely, ensuring patients receive notifications without delays. |
| User Story 2.3: As an optometrist, I want to manage patient appointments efficiently so that I can reduce wait times. | Performance, Maintainability | The scheduling system should handle multiple concurrent bookings and be easy to manage. |
| User Story 3.1: As a diagnosed user, I want my prescription to be saved digitally so that I can easily access it for future use. | Security, Maintainability | Prescriptions should be securely stored and easily retrievable. |
| User Story 3.2: As a low-income patient, I want to check if I qualify for subsidized or donated glasses so that I can afford corrective lenses. | Scalability, Security | The system should handle multiple funding sources while ensuring privacy. |
| **User Story 3.3:** As a patient, I want to track my eyeglasses order status so that I know when they will arrive. | Availability, Performance | The order tracking system should be real-time and always accessible. |
| User Story 4.1: As a non-English-speaking user, I want the platform to support my local language so that I can navigate it easily. | Usability, Accessibility | The platform should be designed with multilingual support and localization. |
| User Story 4.2: As a visually impaired user, I want audio-based guidance in my local language so that I can complete vision tests without reading text. | Usability, Accessibility | The system should provide alternative input/output methods for users with disabilities. |
| User Story 5.1: As a user, I want to track my vision test history so that I can see any changes over time. | Maintainability, Security | The system should securely store test history and ensure it is retrievable when needed. |
| User Story 5.2: As iCare, I want to monitor the number of vision tests completed so that I can measure the platform’s reach and impact. | Scalability, Performance | The system should efficiently handle large volumes of test data and provide analytics. |
| User Story 6.1: As a general user, I want to receive eye health tips and exercises so that I can maintain good vision. | Maintainability, Usability | The recommendation system should be easy to update and provide user-friendly content. |
| User Story 7.1: As iCare, I want to track registered optometrists and caregivers so that I can ensure adequate service coverage. | Scalability, Security | The platform should support a growing network while protecting provider information. |
| User Story 8.1: As a manufacturing partner, I want to receive automated purchase orders based on AI-driven demand forecasting so that I can optimize production schedules. | Performance, Maintainability | The supply chain system should integrate AI and support real-time updates. |
| User Story 9.1: As a lens manufacturer, I want to record and track quality metrics for each batch so that I can maintain high production standards. | Security, Maintainability | The platform should store quality data securely and allow audits. |
| **User Story 10.1:** As a government official, I want to access real-time data on URE prevalence so that I can make informed healthcare policy decisions. | Scalability, Security | The system should support large-scale data collection while ensuring privacy. |
| **User Story 11.1:** As an NPO manager, I want to track the number of beneficiaries served so that I can measure program impact. | Scalability, Security | The platform should track non-profit activities while maintaining data security. |

## 6.2 User Stories grouping based on Quality Attributes

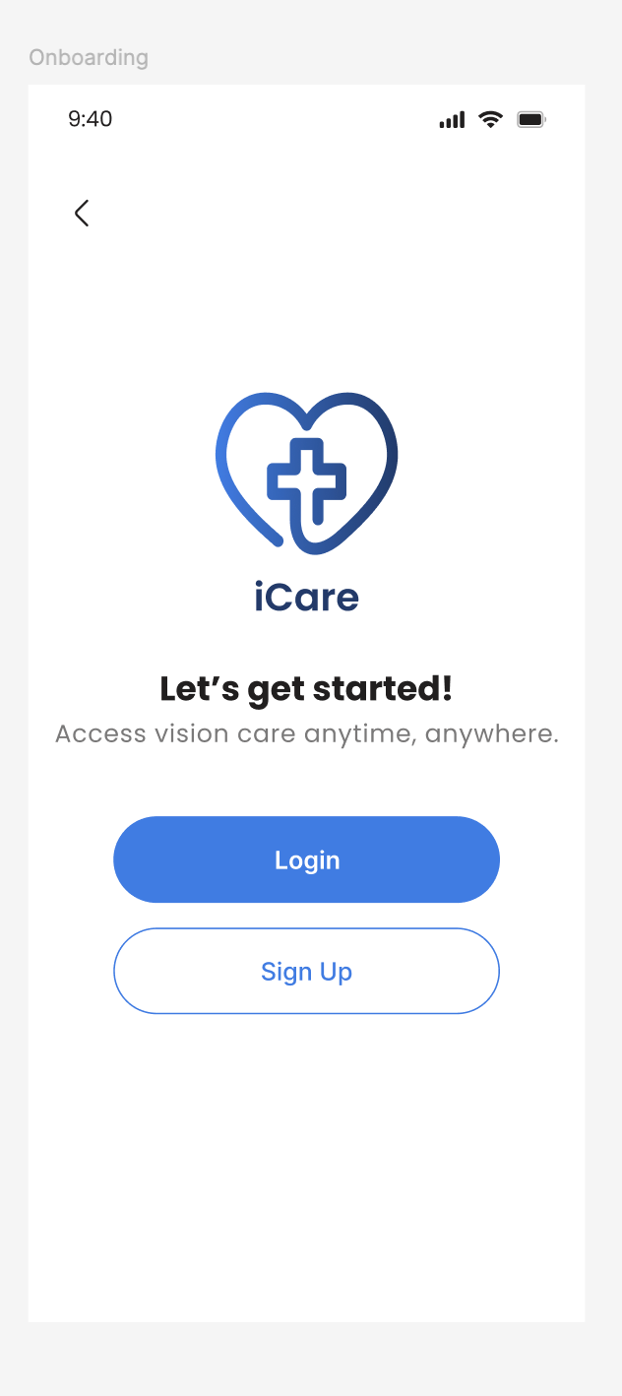
|  |  |  |
| --- | --- | --- |
| Quality attribute | User stories | Explanation of the attributes |
| Availability | 1.1, 1.2, 2.1, 2.2, 3.3 | Ensures the system is accessible anytime for vision testing, appointment booking, and order tracking without downtime |
| Security | 1.2, 1.4, 2.1, 3.1, 3.2, 5.1, 7.1, 9.1, 10.1, 11.1 | Protects sensitive user data such as prescriptions, patient records, funding information, and compliance data. |
| Portability | 1.1, 1.3, 2.1, 4.1, 6.1 | Ensures the system is compatible across different devices (mobile, tablet, desktop) and accessible even in low-bandwidth environments. |
| Performance | 1.1, 2.2, 2.3, 3.3, 5.2, 8.1 | Optimizes system efficiency for fast response times, seamless appointment handling, and AI-driven automation. |
| Usability | 1.1, 1.4, 4.1, 4.2, 6.1 | Ensures the platform is intuitive and accessible, with multilingual support, assistive features, and a user-friendly design. |
| Testability | 3.1, 3.3, 5.1, 6.1, 8.1, 9.1 | Ensures that features like vision tests, prescription management, AI-driven analytics, and compliance tracking can be easily tested and validated. |
| Scalability | 1.3, 3.2, 5.2, 7.1, 8.1, 10.1, 11.1 | Supports growing user demand, increased optometrist networks, and expanding healthcare outreach. |
| Agility | 2.1, 2.3, 7.1, 10.1 | Enables quick adaptation to changing healthcare policies, technology upgrades, and user needs. |
| Supportability | 5.1, 6.1, 7.1, 9.1 | Ensures that the system can be effectively supported and monitored for performance, issue tracking, and future updates |
| Flexibility | 1.3 | Supports various use cases, such as mobile eye camps, rural outreach, and adaptive healthcare solutions. |
| Sustainability | 3.2, 7.1, 8.1, 10.1, 11.1 | Encourages eco-friendly practices, long-term viability, and support for social impact initiatives through nonprofit and government collaborations. |

## User experience and user interface

## 7.1 Value proposition canvas



## 7.2 Create wireframes for important user interactions Login page where user selects preferred language and logs in

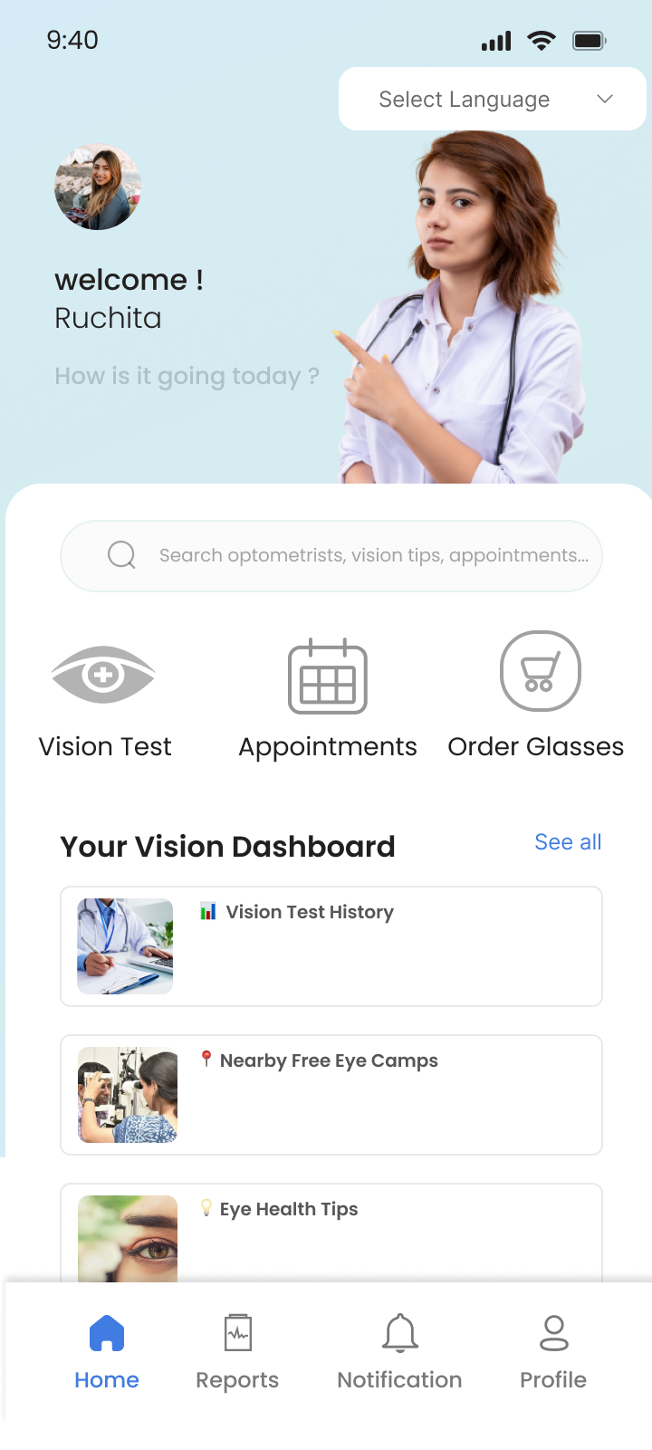
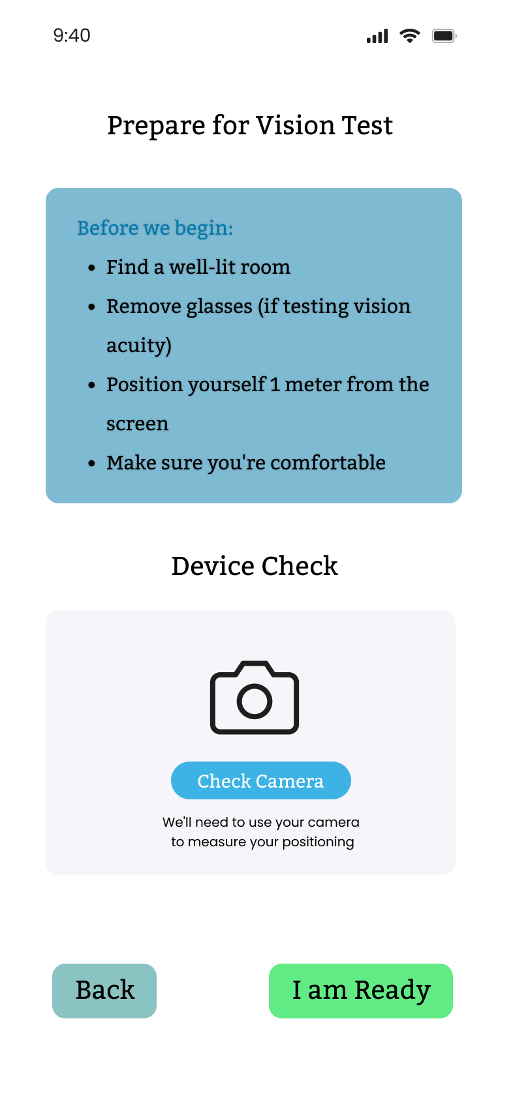


**Logical Sequence for user interactions**

1. Logical Sequence (User) -Self Vision Test

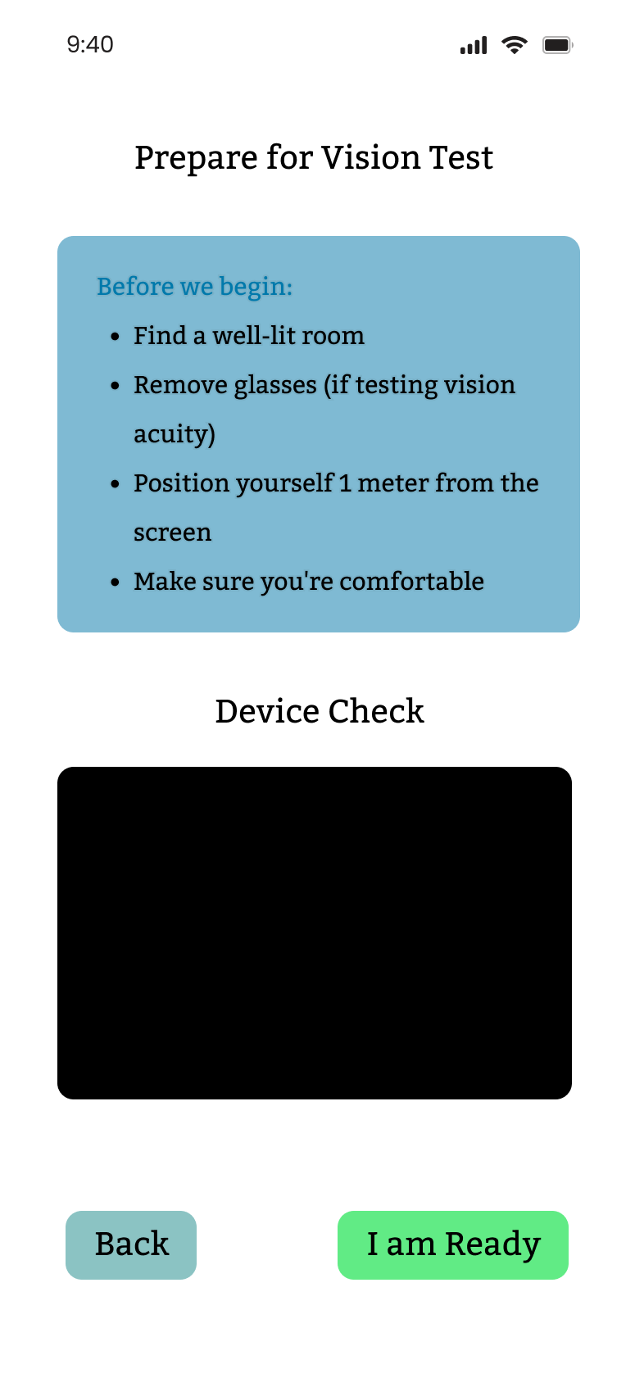
Menu 1. Home screen with menu buttons

* Top Button 1: Vision Test
* Top Button 2: Appointments
* Top Button 3: Order Glasses
* Middle Button 1: Vision Test History
* Middle Button 2: Nearby Free Eye camps
* Middle Button 3: Eye Health Tips
* *Bottom Navigation Buttons:*
* Home – Navigates to the home screen.
* Reports – Opens the reports section where users can view past test results.
* Notification – Shows important alerts and updates.
* Profile – Takes the user to their profile page for personal details and settings.



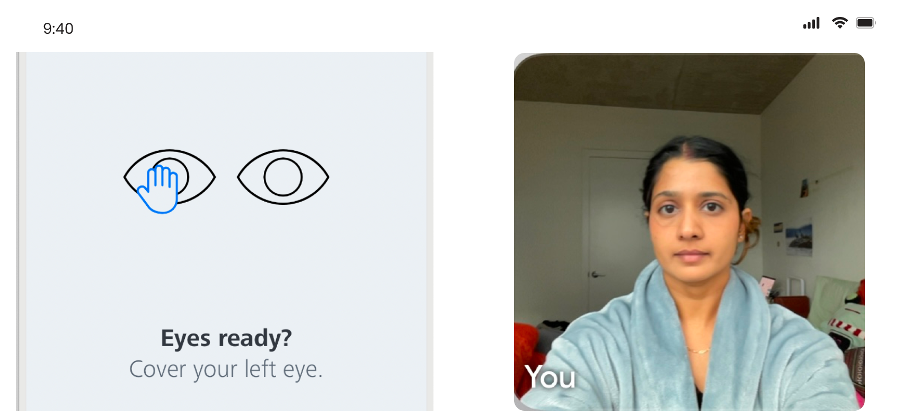
Menu 2: When the user clicks on Vision Test, they are taken to the next page that asks users to prepare for vision test.

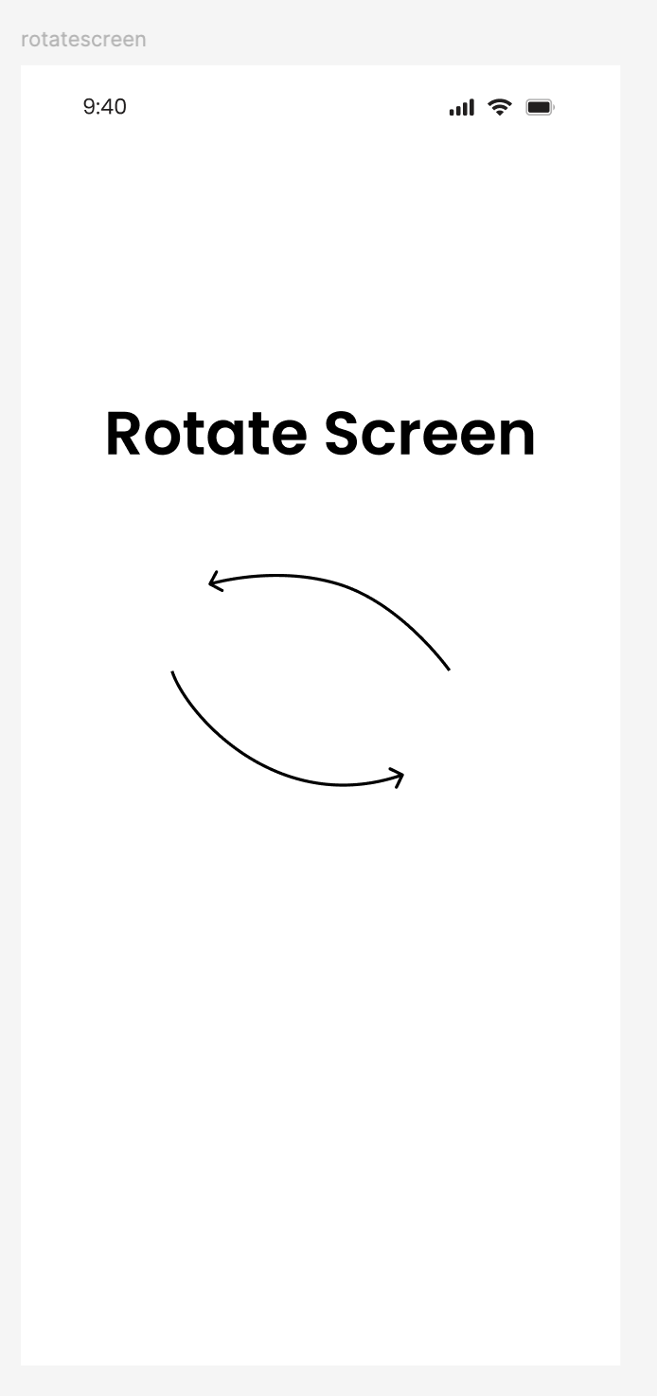
* Button 1: Check Camera
* Button 2: Back
* Button 3: I am Ready

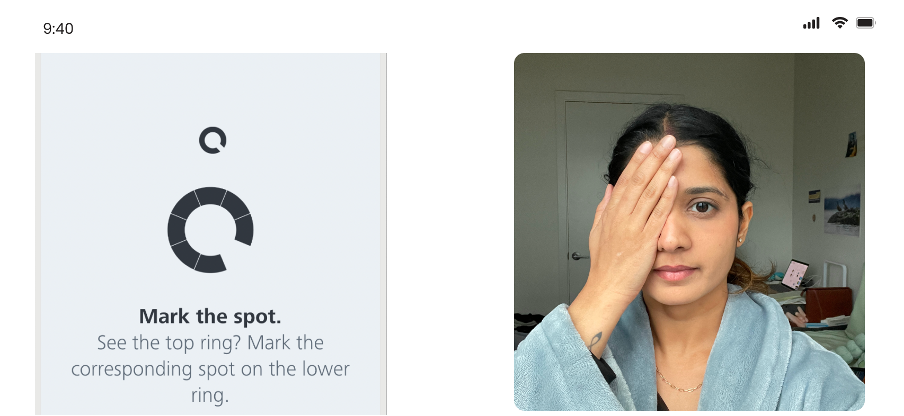


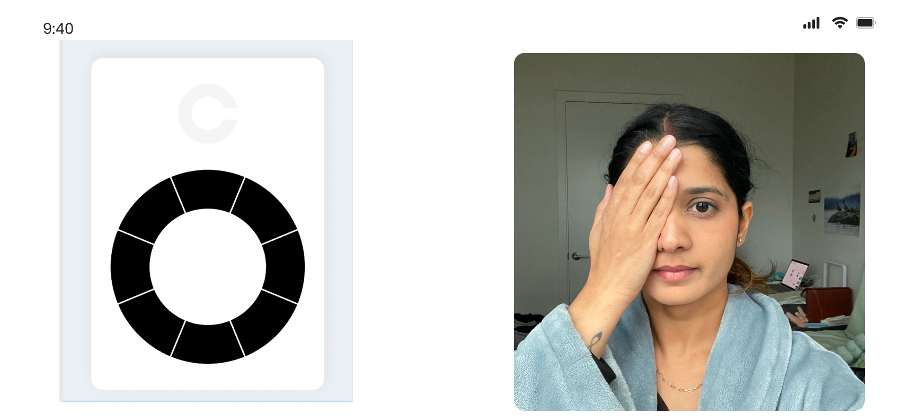
Menu 3: User clicks on Check Camera and Front camera is opened

* Button 1: Back
* Button 2: I am Ready









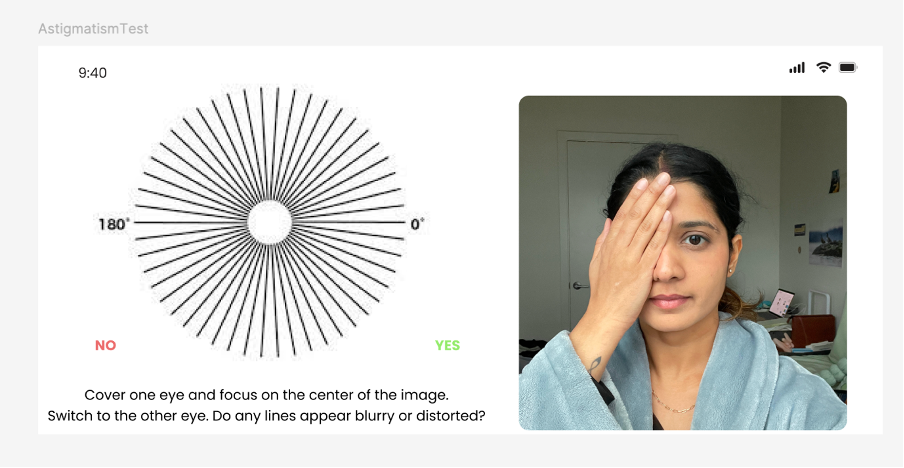
Menu 4: User clicks on I am ready and is asked to rotate the screen. Once the screen is rotated, the user sees the Vision test page. The user is required to identify the location of the gap in the top ring and mark the corresponding spot on the lower ring. This test assesses visual acuity and spatial perception, helping detect issues like astigmatism or other vision impairments. Each button corresponds to a possible location of the gap in the Landolt C ring, and selecting the correct button helps determine the user's visual acuity.

* Top-Right (1 o’clock position)
* Right (3 o’clock position)
* Bottom-Right (5 o’clock position)
* Bottom (6 o’clock position)
* Bottom-Left (7 o’clock position)
* Left (9 o’clock position)
* Top-Left (11 o’clock position)
* Top (12 o’clock position)



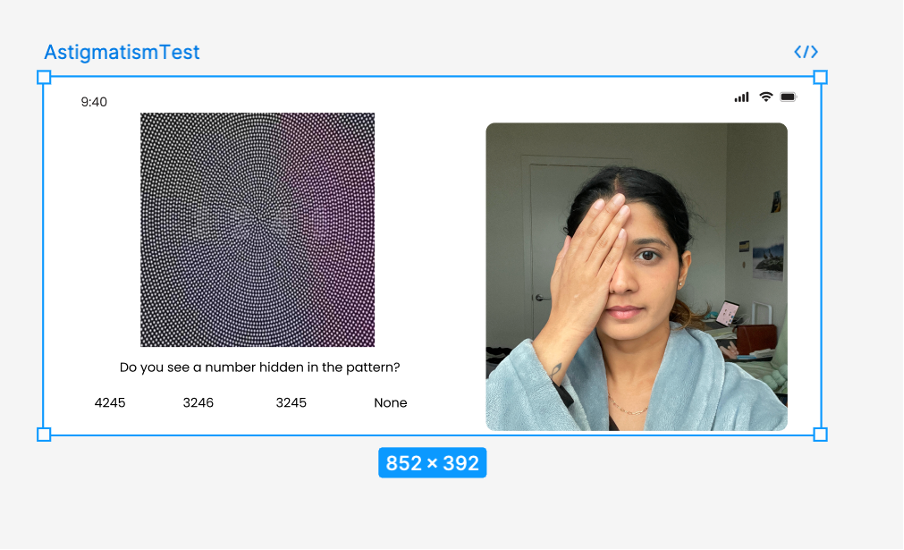
Menu 5: This page is part of a color blindness test where the user is required to identify the hidden shape within the Ishihara color plate. The shape is formed using colored dots, and individuals with color vision deficiencies may struggle to perceive it correctly. The user selects one of the four options to determine potential color blindness types, such as red-green or total color blindness. Here is the list documenting the 4 buttons for this color blindness test page:

* Deer
* Bird
* Butterfly
* Nothing



Menu 6: This page is part of an Astigmatism Test, where the user is required to focus on the radial pattern while covering one eye at a time. If some lines appear blurred, distorted, or darker than others, it may indicate the presence of astigmatism—a common refractive error caused by an irregularly shaped cornea. The test helps assess the visual uniformity and detect potential vision issues. Buttons for Astigmatism Test Page

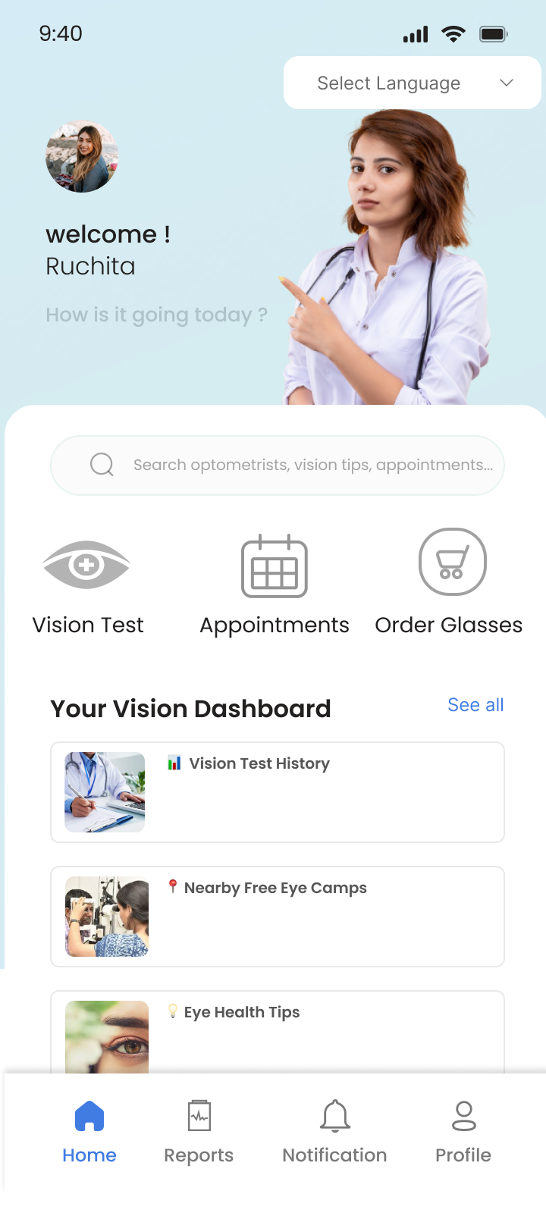
* NO (User does not see any distortion or blurriness in the lines)
* YES (User sees some lines appearing blurry, distorted, or darker than others)



Menu 7: This page is part of a vision test designed to assess contrast sensitivity and detect potential vision impairments, such as color blindness or astigmatism. The user is shown an image with a hidden number embedded within a patterned background. They must select the correct number from the given options or choose "None" if they do not see a number. The results help evaluate the user's ability to distinguish subtle contrasts and detect visual distortions.

Buttons for Hidden Number Vision Test Page

* 4245
* 3246
* 3245
* None (User does not see any number in the pattern)

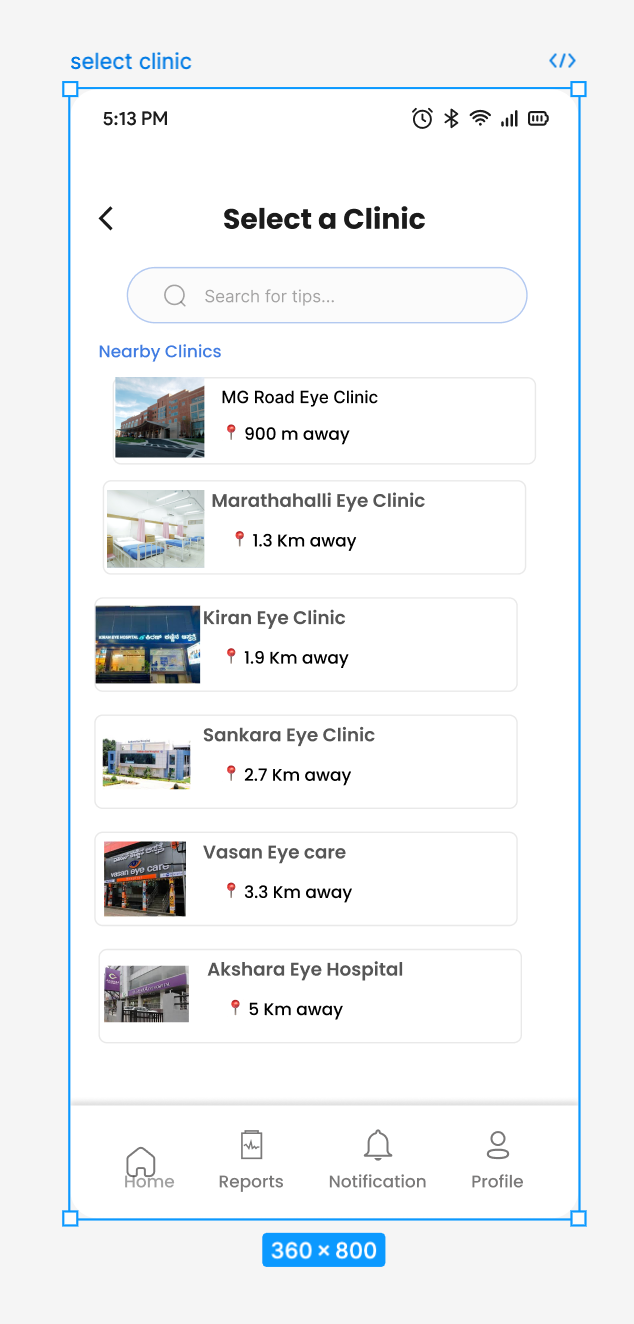


**Logical Sequence (User) - Appointment booking Logical sequence**

Login page where user selects preferred language

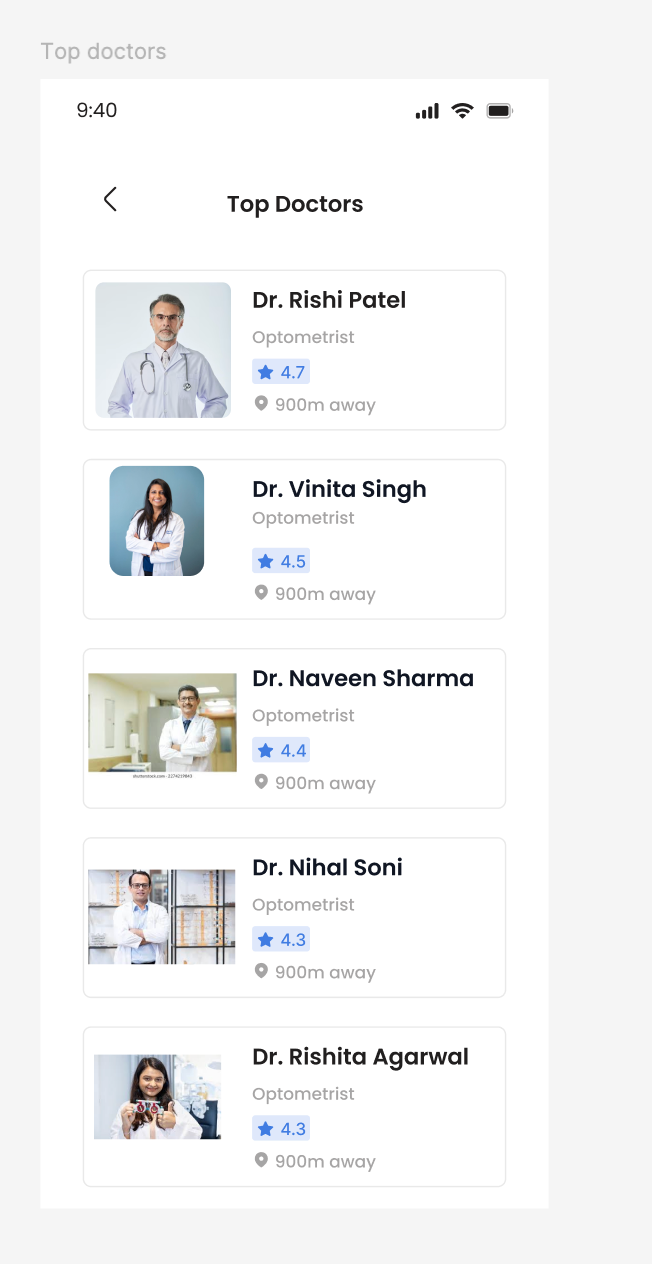
Menu 1. Home screen with menu buttons

* Top Button 1: Vision Test
* Top Button 2: Appointments
* Top Button 3: Order Glasses
* Middle Button 1: Vision Test History
* Middle Button 2: Nearby Free Eye camps
* Middle Button 3: Eye Health Tips
* *Bottom Navigation Buttons:*
* Home – Navigates to the home screen.
* Reports – Opens the reports section where users can view past test results.
* Notification – Shows important alerts and updates.
* Profile – Takes the user to their profile page for personal details and settings.



Menu 2: When the user clicks on the appointments button, a new menu screen opens with a list of nearby hospitals. This page allows users to select a nearby eye clinic for appointments or consultations. At the top, there is a search bar where users can type in the clinic name or location to filter results. Below the search bar, a list of nearby clinics is displayed, showing clinic names, distances from the user, and associated images for easy identification. Each clinic card is interactive and acts as a button to proceed with the selection.

* Back Button (←) – Returns the user to the previous screen.
* Search Bar – Allows users to search for clinics by name or location.
* Clinic Selection Buttons – Each clinic card is a clickable button that allows the user to select a clinic and proceed to the next step.
* *Bottom Navigation Buttons:*
* Home – Navigates to the home screen.
* Reports – Opens the reports section where users can view past test results.
* Notification – Shows important alerts and updates.
* Profile – Takes the user to their profile page for personal details and settings.



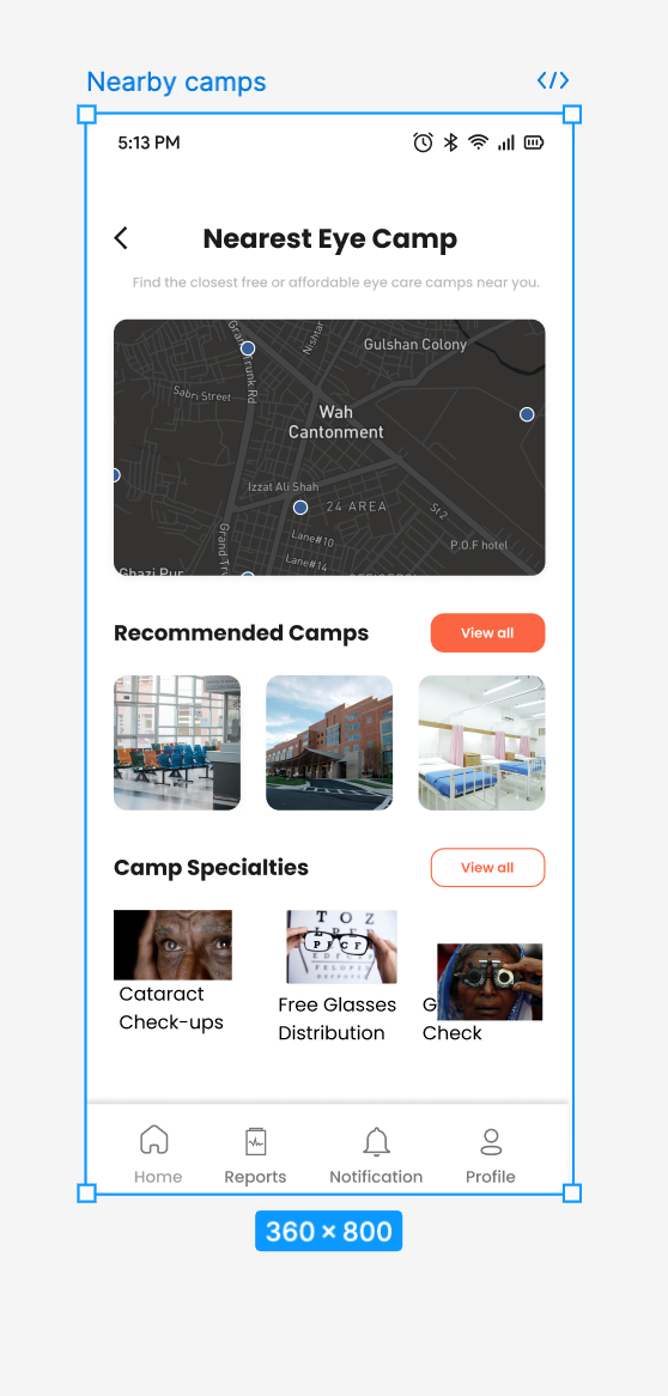
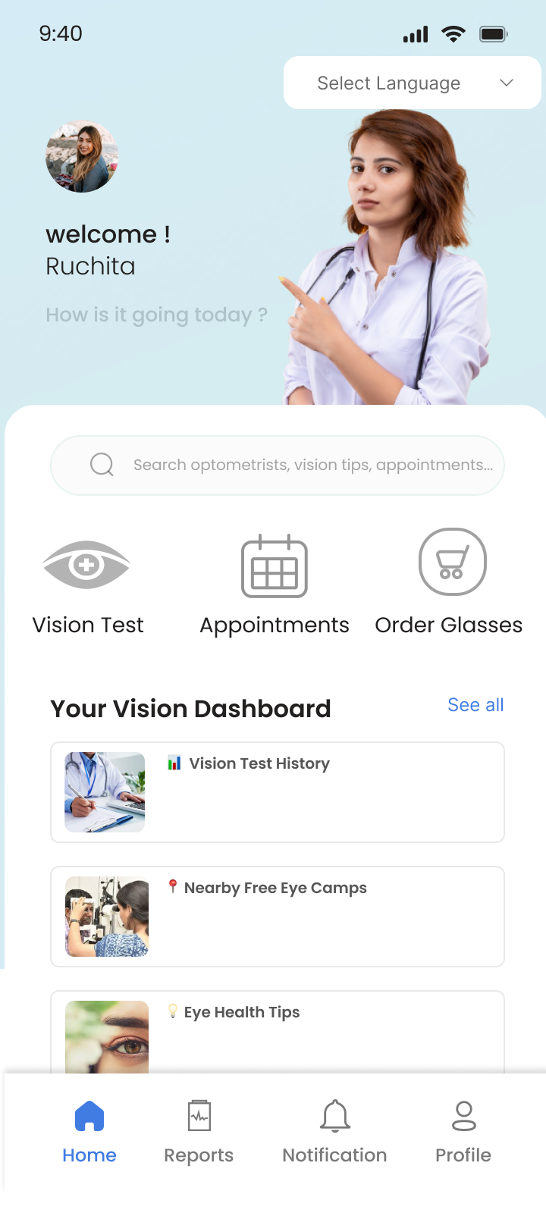
Menu 3: User selects a Clinic and a new page opens where the user sees a list of available optometrists.

* List items: Available doctors



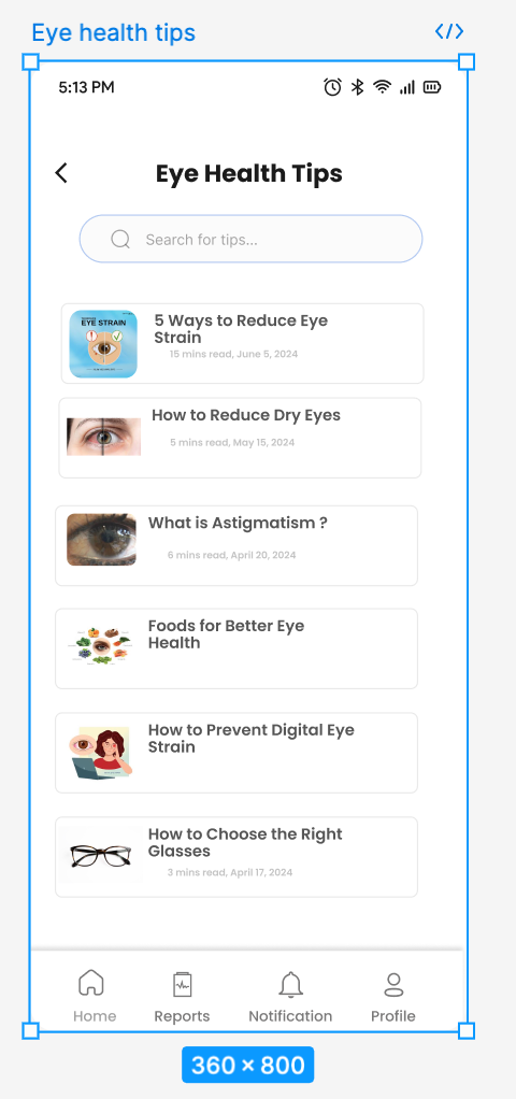
Menu 4: This page allows users to view detailed information about an optometrist and book an appointment. It displays the doctor's profile, clinic location, distance, and ratings. Users can read about the doctor's expertise and available services. Below, a calendar interface enables users to select a preferred date and choose an available time slot. Once a slot is selected, they can confirm their appointment by clicking the "Book Appointment" button.

* Back Button (←) – Navigates back to the previous screen.
* Date Selection Buttons – Allows users to choose an appointment date (e.g., Mon 21, Tue 22, Wed 23, etc.).
* Time Slot Buttons – Enables users to select an available appointment time (e.g., 09:00 AM, 10:00 AM, 02:00 PM, etc.).
* "Book Appointment" Button – Confirms the selected appointment and proceeds with booking.

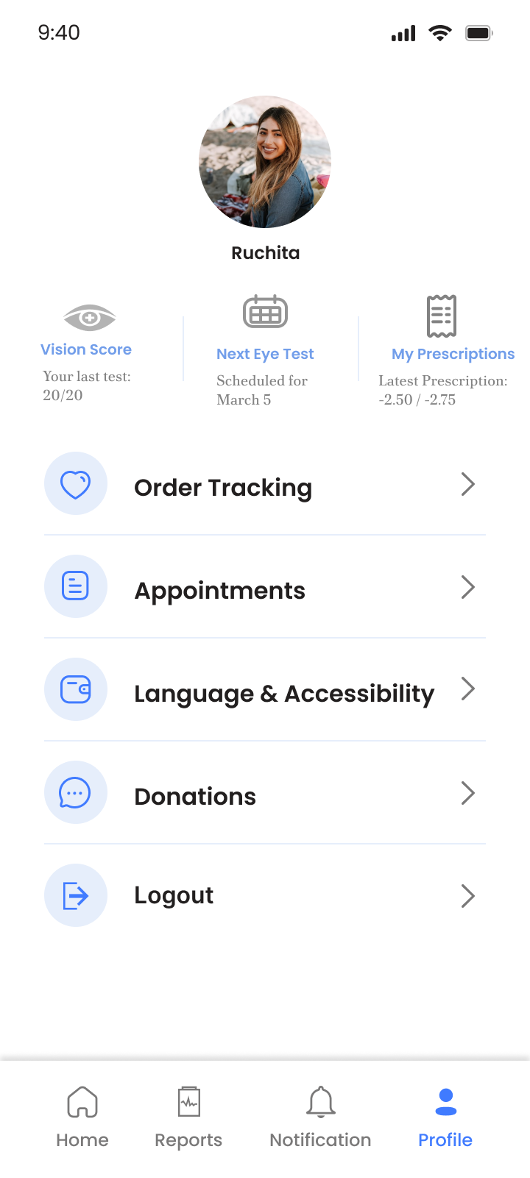


Logical Sequence 3:

User clicks on nearby free eye camps



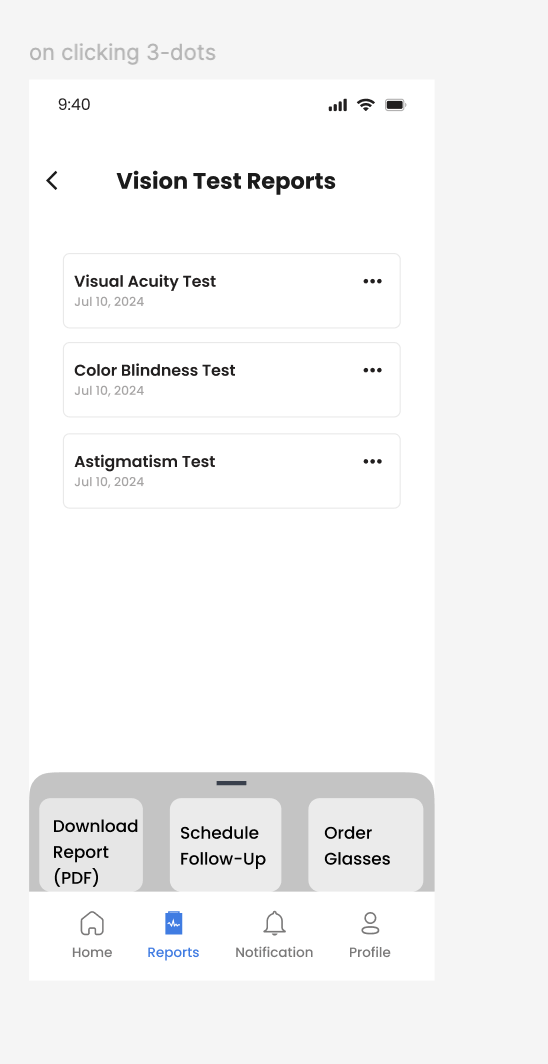
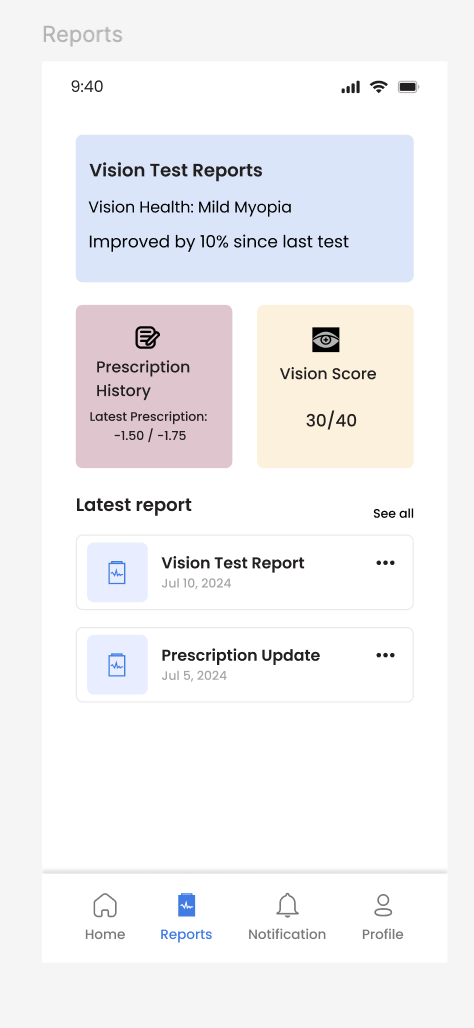
Logical Sequence 4: User clicks on eye health tips

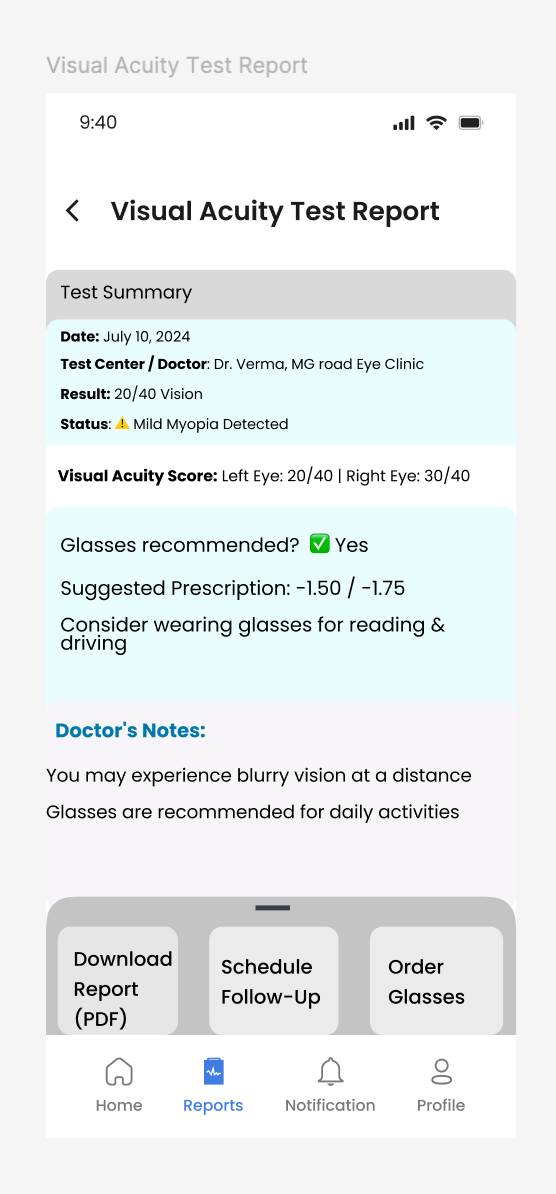
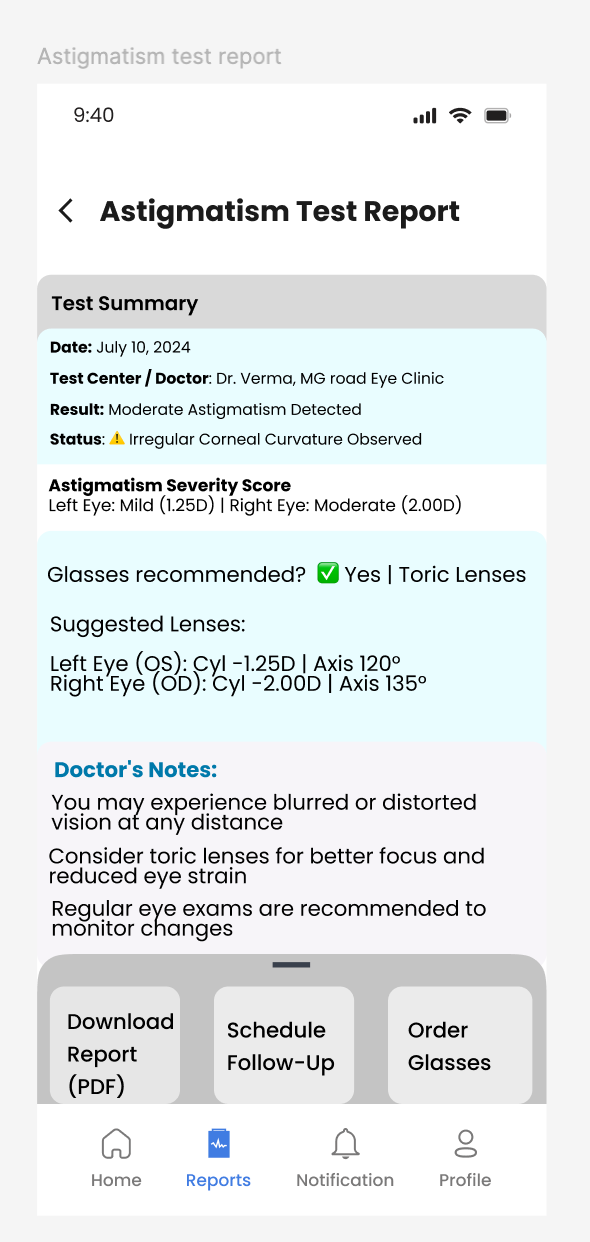
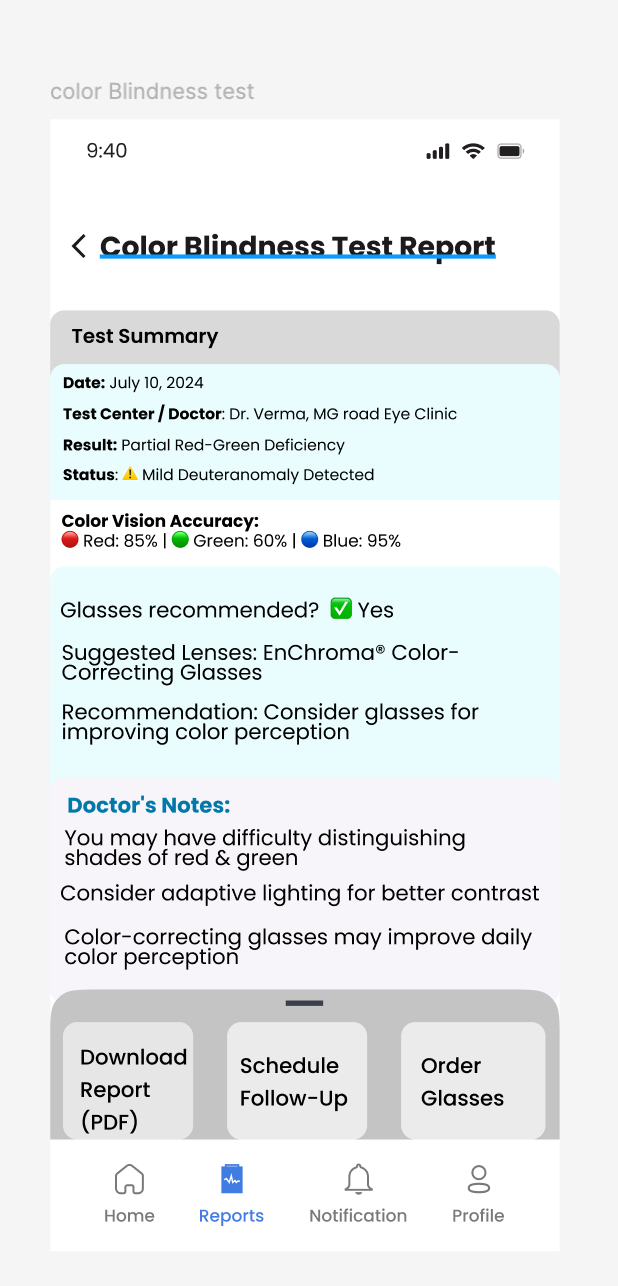


Logical Sequence 5: User clicks on profile

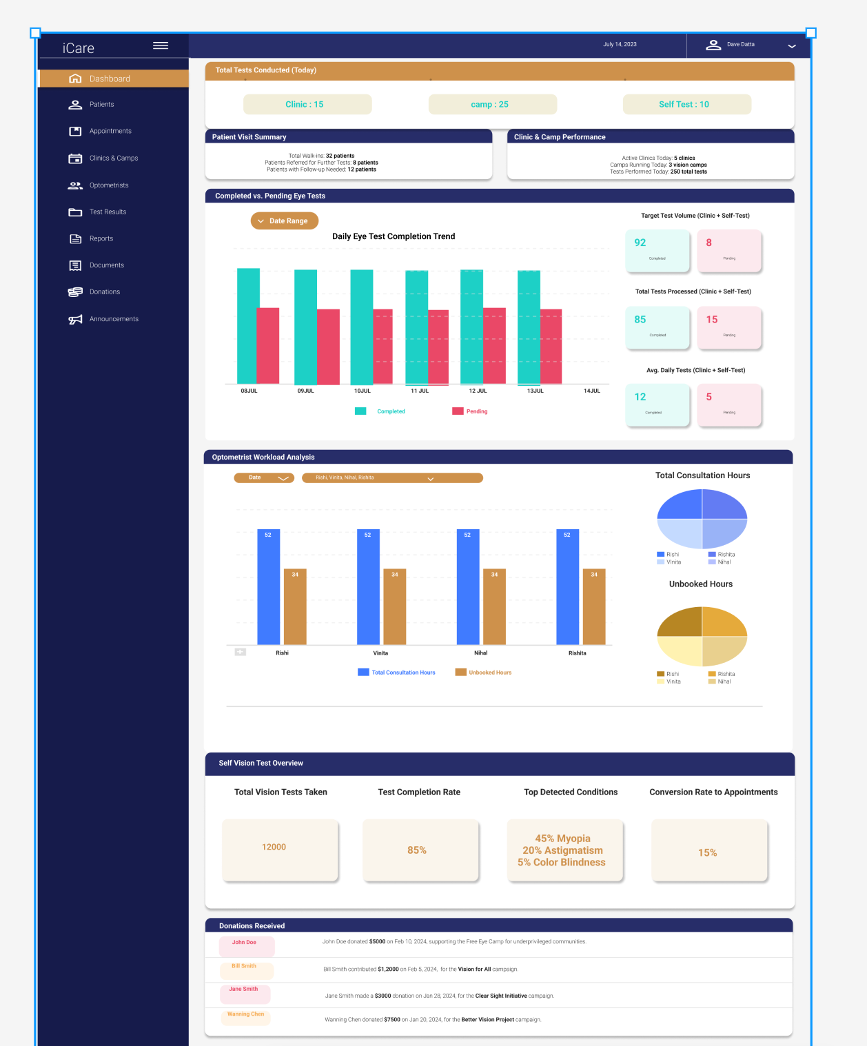


Logical Sequence 6: User clicks on Notifications

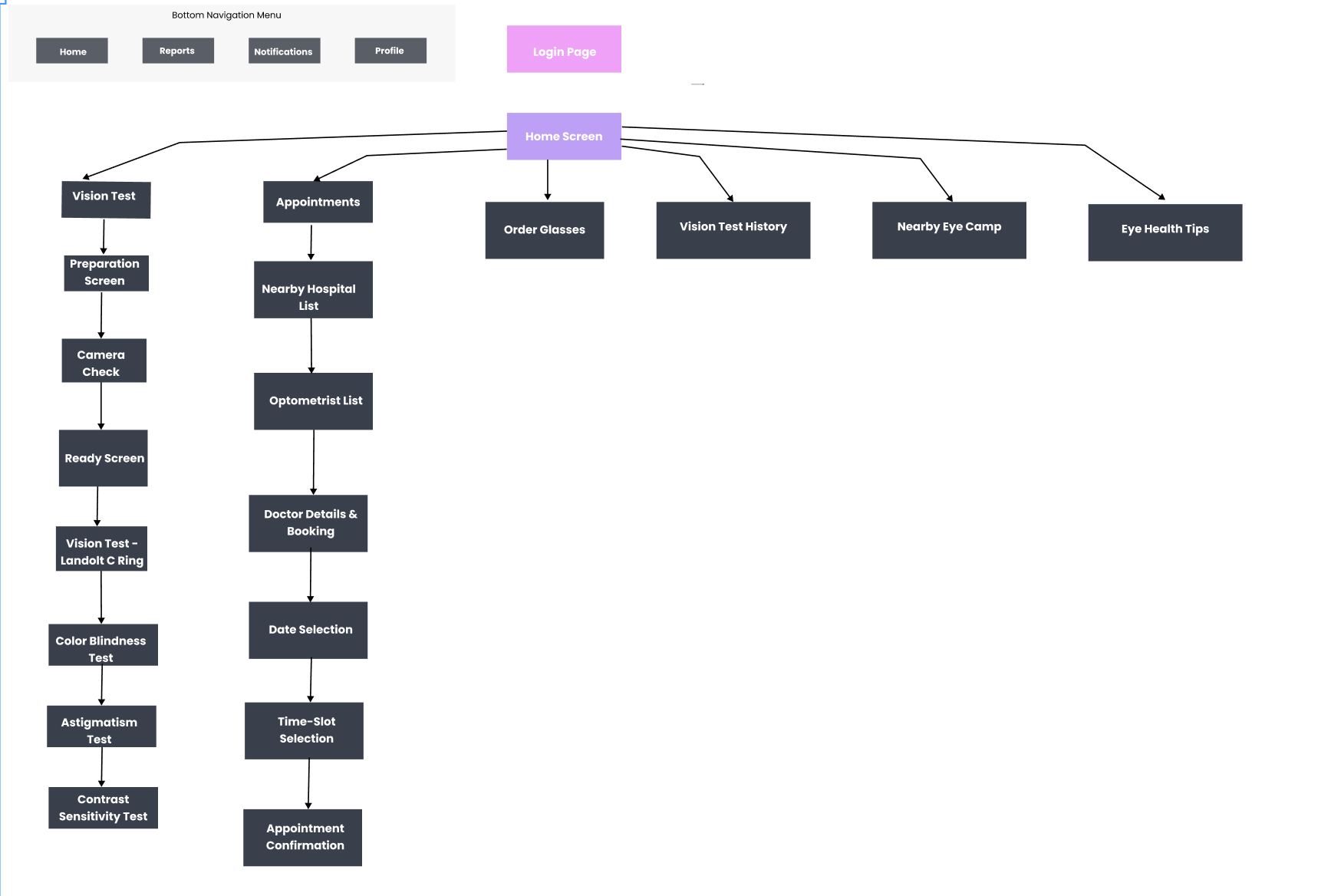




Logical Sequence 7: User clicks on Reports

iCare admin dashboard

## 

UI flow sequence diagram

## 

## 7.3 Build UX/UI architecture

## Information Architecture

## iCare Data Architecture

We have categorized iCare data based on their functional areas and application development strategies. This data architecture is created to ensure efficient data management, optimized query operations and support scalability. We have created 5-NF tables for current data and denormalized tables for historical data for faster data retrieval, reliability and scalability.

The data model is divided into the following core categories:

1. User management - Manages Users, Authentication, Preferences, and access roles.
2. Healthcare & Appointments - Tracks patient visits, provider availability, vision tests and prescriptions.
3. Organization management - Unifies external entities such as schools, suppliers, non-profits under a standardized structure.
4. Camp & Location - Manages healthcare camps, schedules and operational statistics.
5. Supply Chain & Inventory - Manages inventory, suppliers, stock management and shipment tracking.
6. Donations & Fundraising - Handles donor information, fundraising campaigns & donation allocations.
7. Historical tables - Stores precomputed reports and analytical data for efficient reporting.

## User Management Tables:

### Master Data Tables:

1. User\_Profile: Stores basic user details

| Table: User\_Profile (Field | Data Type | Constraint) | | |
| --- | --- | --- |
| User\_Id | UUID | Primary Key |
| First\_name | STRING | NOT NULL |
| Last\_name | STRING | NOT NULL |
| Middle\_name | STRING |  |
| Email | VARCHAR | UNIQUE |
| Age | INT | NOT NULL |
| Gender\_Id | INT | Ref. Gender (Gender\_Id) |
| Communication\_language\_Id | INT | Ref. Language (Language\_Id) |
| User\_Type\_Id | INT | Ref. User\_Type (User\_Type\_Id) |
| Last\_update\_dtm | DATETIME | NOT NULL |
| Last\_update\_user\_Id | UUID | NOT NULL |

1. User\_Preferences: Stores user-specific settings

| Table: User\_Preferences (Field | Data Type | Constraint) | | |
| --- | --- | --- |
| Preference\_Id | UUID | Primary Key |
| User\_Id | UUID | Ref. User\_Profile (User\_Id) |
| Consent\_guardian\_data\_access | BOOLEAN |  |
| Consent\_research\_data\_access | BOOLEAN |  |
| Appointment\_reminder | BOOLEAN |  |
| Vision\_Test\_reminder | BOOLEAN |  |
| Notification\_Type\_Id | INT | Ref. Notification\_Type (Notification\_Id) |

1. User\_Authentication: Tracks multiple authentication methods per user

| Table: User\_Authentication (Field | Data Type | Constraint) | | |
| --- | --- | --- |
| Auth\_Id | UUID | Primary Key |
| User\_Id | UUID | Ref. User\_Profile (User\_Id) |
| Auth\_Type\_Id | INT | Ref. Authentication\_Type (Auth\_Type\_Id) |
| Auth\_Identifier | VARCHAR | UNIQUE NOT NULL |
| Password\_hash | VARCHAR | NULLABLE (NULL for social logins) |
| Last\_update\_dtm | DATETIME | NOT NULL |
| Last\_update\_user\_Id | UUID | NOT NULL |

1. User\_Address: Stores multiple addresses per user

| Table: User\_Address (Field | Data Type | Constraint) | | |
| --- | --- | --- |
| Address\_Id | UUID | Primary Key |
| User\_Id | UUID | Ref. User\_Profile (User\_Id) |
| Address\_Type\_Id | INT | Ref. Address\_Type (Address\_Type\_Id) |
| Address\_Line\_1 | VARCHAR | NOT NULL |
| Address\_Line\_2 | VARCHAR |  |
| City | VARCHAR | NOT NULL |
| State | VARCHAR | NOT NULL |
| Country | VARCHAR | NOT NULL |
| Zip | VARCHAR | NOT NULL |
| Last\_update\_dtm | DATETIME | NOT NULL |
| Last\_update\_user\_Id | UUID | NOT NULL |

1. User\_Role\_Association: Associates users with multiple roles

| Table: User\_Role\_Association (Field | Data Type | Constraint) | | |
| --- | --- | --- |
| Association\_Id | UUID | Primary Key |
| User\_Id | UUID | Ref. User\_Profile (User\_Id) |
| Role\_Id | INT | Ref. User\_Role (User\_role\_Id) |
| Last\_update\_dtm | DATETIME | NOT NULL |
| Last\_update\_user\_Id | UUID | NOT NULL |

1. User\_Phone: Stores multiple phone numbers per user

| Table: User\_Phone (Field | Data Type | Constraint) | | |
| --- | --- | --- |
| Phone\_Id | UUID | Primary Key |
| User\_Id | UUID | Ref. User\_Profile (User\_Id) |
| Phone\_Type\_Id | INT | Ref. Phone\_Type (Phone\_Type\_Id) |
| Phone\_number | VARCHAR | UNIQUE NOT NULL |
| Is\_Primary | BOOLEAN |  |
| Last\_update\_dtm | DATETIME | NOT NULL |
| Last\_update\_user\_Id | UUID | NOT NULL |

### Lookup Tables:

1. User\_Type : (User\_Type\_Id: INT, User\_Type\_Name: VARCHAR)
   * Values: Individual, Organization, Donor, etc.
2. Gender : (Gender\_Id: INT, Gender\_Name: VARCHAR)
   * Values: Male, Female, Transgender, etc.
3. Address\_Type : (Address\_Type\_Id: INT, Address\_Type\_Name: VARCHAR)
   * Values: Home, Office, etc.
4. Language : (Language\_Id: INT, Language\_Name: VARCHAR)
   * Values: Hindi, Tamil, English, etc.
5. Phone\_Type : (Phone\_Type\_Id: INT, Phone\_Type\_Name: VARCHAR)
   * Values: Home, Mobile, Work etc.
6. Authentication\_Type : (Auth\_Type\_Id: INT, Auth\_Type\_Name: VARCHAR)
   * Values: Password, Google, Phone, etc.
7. Notification\_Type : (Notification\_Type\_Id: INT, Notification\_Type\_Name: VARCHAR)
   * Values: Email, SMS, Phone, etc.
8. Communication\_Channel : (Communication\_Channel\_Id: INT, Communication\_Channel\_Name: VARCHAR)
   * Values: WhatsApp, Mobile, SMS, etc.
9. Accessibility\_Mode : (Mode\_Id: INT, Mode\_Name: VARCHAR)
   * Values: Screen Reader, High Contrast, etc.
10. Font\_Size : (Font\_Size\_Id: INT, Font\_Size\_Name: VARCHAR)
    * Values: Small, Medium, Large, etc.
11. User\_Theme : (User\_Theme\_Id: INT, User\_Theme\_Name: VARCHAR)
    * Values: Light, Dark, Auto, etc.
12. User\_Role : (User\_Role\_Id: INT, User\_Role\_Name: VARCHAR)
    * Values: Patient, Optometrist, Caregiver, Admin, Student, Teacher etc

## Organization Management Tables:

### Master Data Tables:

1. Organization: Tracks organizations like Schools, NPOs, Clinic etc

| Table: Organization\_Profile (Field | Data Type | Constraint) | | |
| --- | --- | --- |
| Organization\_Id | UUID | Primary Key |
| Organization\_name | VARCHAR | NOT NULL |
| Organization\_Type\_Id | INT | Ref. Organization\_Type (Org\_Type\_Id) |
| Organization\_email | VARCHAR | NOT NULL | UNIQUE |
| Last\_update\_dtm | DATETIME | NOT NULL |
| Last\_update\_user\_Id | UUID | NOT NULL |

1. Organization\_Address: Stores multiple addresses per organization

| Table: Organization\_Address (Field | Data Type | Constraint) | | |
| --- | --- | --- |
| Address\_Id | UUID | Primary Key |
| Org\_Id | UUID | Ref. Organization\_Profile (Organization\_Id) |
| Address\_Type\_Id | INT | Ref. Address\_Type (Address\_Type\_Id) |
| Address\_Line\_1 | VARCHAR | NOT NULL |
| Address\_Line\_2 | VARCHAR |  |
| City | VARCHAR | NOT NULL |
| State | VARCHAR | NOT NULL |
| Country | VARCHAR | NOT NULL |
| Zip | VARCHAR | NOT NULL |
| Last\_update\_dtm | DATETIME | NOT NULL |
| Last\_update\_user\_Id | UUID | NOT NULL |

1. Organization\_Phone: Stores multiple phone numbers per organization

| Table: Organization\_Phone (Field | Data Type | Constraint) | | |
| --- | --- | --- |
| Phone\_Id | UUID | Primary Key |
| Organization\_Id | UUID | Ref. Organization\_Profile (Organization\_Id) |
| Phone\_Type\_Id | INT | Ref. Phone\_Type (Phone\_Type\_Id) |
| Phone\_number | VARCHAR | UNIQUE NOT NULL |
| Is\_Primary | BOOLEAN |  |
| Last\_update\_dtm | DATETIME | NOT NULL |
| Last\_update\_user\_Id | UUID | NOT NULL |

1. User\_Org\_Association: Tracks users and their roles in different organizations

| Table: User\_Org\_Association (Field | Data Type | Constraint) | | |
| --- | --- | --- |
| Relationship\_Id | UUID | Primary Key |
| User\_Id | UUID | Ref. User\_Profile (User\_Id) |
| Organization\_Id | UUID | Ref. Organization\_Profile (Organization\_Id) |
| Role\_Id | INT | Ref. User\_Role (User\_Role\_Id) |
| Last\_update\_dtm | DATETIME | NOT NULL |
| Last\_update\_user\_Id | UUID | NOT NULL |

1. User\_Access\_Association: Tracks group of users (Teachers, volunteers) that can access hierarchical user data (students, beneficiaries).

| Table: User\_Access\_Association (Field | Data Type | Constraint) | | |
| --- | --- | --- |
| Access\_Id | UUID | Primary Key |
| Accessor\_Id | UUID | Ref. User\_Profile (User\_Id) |
| Target\_User\_Id | UUID | Ref. User\_Profile (User\_Id) |
| Role\_Id | INT | Ref. User\_Role (User\_Role\_Id) |
| Access\_Type\_Id | INT | Ref. Access\_Type (Access\_Type\_Id) |
| Last\_update\_user\_Id | UUID | NOT NULL |
| Last\_update\_dtm | DATETIME | NOT NULL |

### Lookup Tables:

1. Organization\_Type : (Org\_Type\_Id: INT, User\_Type\_Name: VARCHAR)
   * Values: School, NPO, etc.
2. Access\_Type : (Access\_Type\_Id : INT, Access\_Type\_Name: VARCHAR)
   * Values: Read, Write, etc.

## Healthcare & Appointment Tables:

### Master Data Tables:

1. Healthcare\_provider: Links provider to clinics/ hospitals.

| Table: Healthcare\_Provider (Field | Data Type | Constraint) | | |
| --- | --- | --- |
| Provider\_Id | UUID | PRIMARY KEY | Ref. User\_Profile (User\_Id) |
| Clinic\_Id | UUID | Ref. Organization (Organization\_Id) |
| Last\_update\_user\_Id | UUID | NOT NULL |
| Last\_update\_dtm | DATETIME | NOT NULL |

1. Healthcare\_provider\_speciality: Links provider to specialities.

| Table: Healthcare\_Provider\_Speciality(Field | Data Type | Constraint) | | |
| --- | --- | --- |
| Speciality\_Id | UUID | PRIMARY KEY |
| User\_Id | UUID | Ref. User\_Profile (User\_Id) |
| Provider\_Type\_Id | UUID | Ref. Provider\_Type (Provider\_Type\_Id) |
| Last\_update\_user\_Id | UUID | NOT NULL |
| Last\_update\_dtm | DATETIME | NOT NULL |

1. Healthcare\_data: Data for healthcare and vision educational data.

| Table: Healthcare\_Awareness(Field | Data Type | Constraint) | | |
| --- | --- | --- |
| Topic\_id | UUID | PRIMARY KEY |
| Category | VARCHAR |  |
| Content | JSON |  |
| Last\_update\_user\_Id | UUID | NOT NULL |
| Last\_update\_dtm | DATETIME | NOT NULL |

1. Healthcare\_provider\_availability: Tracks provider schedules.

| Table: Healthcare\_Provider\_Availability(Field | Data Type | Constraint) | | |
| --- | --- | --- |
| Availability\_Id | UUID | PRIMARY KEY |
| Provider\_Id | UUID | Ref. Healthcare\_provider (Provider\_Id) |
| Date | DATE | NOT NULL |
| Start\_Time | TIME | NOT NULL |
| End\_Time | TIME | NOT NULL |
| Status\_Id | INT | Ref. Availability\_Status (Status\_Id) |
| Last\_update\_user\_Id | UUID | NOT NULL |
| Last\_update\_dtm | DATETIME | NOT NULL |

1. Appointments: Tracks appointments between patients & providers.

| Table: Appointments (Field | Data Type | Constraint) | | |
| --- | --- | --- |
| Appointment\_Id | BIGINT | PRIMARY KEY |
| Patient\_Id | UUID | Ref. User\_Profile (User\_Id) |
| Provider\_Id | UUID | Ref. Healthcare\_provider (Provider\_Id) |
| Appointment\_Date | DATE | NOT NULL |
| Provider\_Availability\_Id | UUID | Ref. Healthcare\_provider\_availability (Availability\_Id) |
| Status\_Id | INT | Ref. Appointment\_Status (Status\_Id) |
| Last\_update\_user\_Id | UUID | NOT NULL |
| Last\_update\_dtm | DATETIME | NOT NULL |

1. Vision\_Test: Tracks patient eye tests.

| Table: Vision\_Test (Field | Data Type | Constraint) | | |
| --- | --- | --- |
| Test\_Id | UUID | PRIMARY KEY |
| Patient\_Id | UUID | Ref. User\_Profile (User\_Id) |
| Provider\_Id | UUID | Ref. Healthcare\_provider (Provider\_Id) |
| Secondary\_provider\_Id | UUID | Ref. Healthcare\_provider (Provider\_Id) |
| Test\_date | DATE | NOT NULL |
| Appointment\_Id | BIGINT | Ref. Appointments (Appointment\_Id) |
| Eye\_Distance | FLOAT | NOT NULL |
| Diagnosis\_Note | VARCHAR | NULLABLE |
| Last\_update\_user\_Id | UUID | NOT NULL |
| Last\_update\_dtm | DATETIME | NOT NULL |

1. Vision\_Test\_Details: Stores test parameters.

| Table: Vision\_Test\_Details (Field | Data Type | Constraint) | | |
| --- | --- | --- |
| Detail\_Id | UUID | PRIMARY KEY |
| Test\_Id | UUID | Ref. Vision\_Test (Test\_Id) |
| Parameter\_Id | INT | Ref. Test\_Parameter (Parameter\_Id) |
| Parameter\_value | VARCHAR | NOT NULL |
| Last\_update\_user\_Id | UUID | NOT NULL |
| Last\_update\_dtm | DATETIME | NOT NULL |

1. Prescription: stores prescriptions.

| Table: Prescription (Field | Data Type | Constraint) | | |
| --- | --- | --- |
| Prescription\_Id | UUID | PRIMARY KEY |
| Patient\_Id | UUID | Ref. User\_details (User\_Id) |
| Test\_Id | UUID | Ref. Vision\_Test (Test\_Id) |
| Provider\_Id | UUID | Ref. Healthcare\_provider (provider\_Id) |
| Prescription\_Date | DATE | NOT NULL |
| Expiry\_Date | DATE | NOT NULL |
| Last\_update\_user\_Id | UUID | NOT NULL |
| Last\_update\_dtm | DATETIME | NOT NULL |

1. Prescription\_Details: stores eye prescription data.

| Table: Prescription\_Details (Field | Data Type | Constraint) | | |
| --- | --- | --- |
| Detail\_Id | UUID | Primary key |
| Prescription\_Id | UUID | Ref. Prescription (Prescription\_Id) |
| Eye\_side\_Id | INT | Ref. Eye\_Side (Eye\_side\_Id) |
| Sphere | FLOAT | NULLABLE |
| Cylinder | FLOAT | NULLABLE |
| Axis | INT | NULLABLE |
| Addition | FLOAT | NULLABLE |
| Last\_update\_user\_Id | UUID | NOT NULL |
| Last\_update\_dtm | DATETIME | NOT NULL |

### Lookup Tables:

1. Provider\_Type : (Provider\_Type\_Id: INT, Provider\_Type\_Name: VARCHAR)
   1. Values: Therapist, Caregiver, Ophthalmologist, etc.
2. Appointment\_Status: (Appointment\_Status\_Id: INT, Appointment\_Status\_Name: VARCHAR)
   1. Values: Scheduled, Canceled, Completed, etc.
3. Availability\_Status: (Availability\_Status\_Id: INT, Availability\_Status\_Name: VARCHAR)
   1. Values: Available, Booked, Not Available, etc.
4. Test\_Parameter: (Test\_parameter\_Id: INT, Test\_parameter\_name: VARCHAR)
   1. Values: Visual Acuity, Color Vision, etc.
5. Eye\_Side: (Eye\_side\_Id: INT, Eye\_side\_name: VARCHAR)
   1. Values: Left, Right, Both, etc.

## Camp & Location Tables:

### Master Data Tables:

1. Camp\_Location: Master table for all service locations.

| Table: Camp\_Location (Field | Data Type | Constraint) | | |
| --- | --- | --- |
| Camp\_Location\_Id | UUID | Primary Key |
| Name | VARCHAR | NOT NULL |
| Latitude | DECIMAL (9,6) | NOT NULL |
| Longitude | DECIMAL (9,6) | NOT NULL |
| Address | VARCHAR | NOT NULL |
| State | VARCHAR | NOT NULL |
| Country | VARCHAR | NOT NULL |
| Zip | VARCHAR | NOT NULL |
| Location\_Type\_Id | INT | Ref. Location\_Type (Location\_Type\_Id) |

1. Camp: Stores information about specific camps at different locations.

| Table: Camp (Field | Data Type | Constraint) | | |
| --- | --- | --- |
| Camp\_Id | UUID | Primary Key |
| Camp\_Type\_Id | INT | Ref. Camp\_Type (Camp\_Type\_Id) |
| Camp\_Location\_Id | UUID | Ref. Camp\_Location (Camp\_location\_Id) |
| Planned\_daily\_capacity | INT | NOT NULL |
| Planned\_daily\_appointments | INT | NOT NULL |
| Planned\_daily\_doctors | INT | NOT NULL |
| Start\_Date | DATE | NOT NULL |
| End\_Date | DATE | NOT NULL |
| Npo\_Id | UUID | Ref. Organization (Organization\_Id) |

1. Camp\_Daily\_Schedule: Stores daily camp operating hours & statistics.

| Table: Camp\_Daily\_Schedule (Field | Data Type | Constraint) | | |
| --- | --- | --- |
| Schedule\_Id | UUID | Primary Key |
| Camp\_Id | UUID | Ref. Camp (Camp\_Id) |
| Record\_Date | DATE | NOT NULL |
| Start\_Time | TIME | NOT NULL |
| End\_Time | INT | NOT NULL |
| Number\_Capacity | INT | NOT NULL |
| Number\_Appointments | INT | NOT NULL |
| Number\_Doctors | INT | NOT NULL |
| Patients\_Attended | INT | NOT NULL |
| Medications\_Dispensed | INT | NOT NULL |
| Volunteers\_Present | INT | NOT NULL |
| Last\_update\_user\_Id | UUID | NOT NULL |
| Last\_update\_dtm | DATETIME | NOT NULL |

### Lookup Tables:

1. Location\_Type : (Location\_Type\_Id: INT, Location\_Type\_Name: VARCHAR)
   1. Values: Fixed Clinic, Mobile Camp, Temporary Site, etc.

2. Camp\_Type : (Camp\_Type\_Id: INT, Camp\_Type\_Name: VARCHAR)

* 1. Values: Testing, Distribution, Comprehensive, etc.

## Inventory & Shipping Tables:

### Master Data Tables:

1. Inventory: Stores product required for iCare operations.

| Table: Inventory (Field | Data Type | Constraint) | | |
| --- | --- | --- |
| Inventory\_Id | UUID | Primary Key |
| Product\_Name | VARCHAR | NOT NULL |
| Supplier\_Id | UUID | Ref. Supplier (Supplier\_Id) |
| Stock\_Quantity | INT | NOT NULL |
| Last\_update\_user\_Id | UUID | NOT NULL |
| Last\_update\_dtm | DATETIME | NOT NULL |

1. Supplier: Stores Supplier information for sourcing inventory items.

| Table: Supplier (Field | Data Type | Constraint) | | |
| --- | --- | --- |
| Supplier\_Id | UUID | Primary Key  Ref. Organization (Organization Id) |
| Supplier\_Name | VARCHAR | NOT NULL |
| Supplier\_Status | INT | Ref. Supplier\_Compliance (Compliance\_Type\_Id) |
| Last\_update\_user\_Id | UUID | NOT NULL |
| Last\_update\_dtm | DATETIME | NOT NULL |

1. Inventory\_stock: Tracks stock levels for each inventory item.

| Table: Inventory\_Stock (Field | Data Type | Constraint) | | |
| --- | --- | --- |
| Stock\_Id | UUID | Primary Key |
| Inventory\_Id | UUID | Ref. Inventory (Inventory\_Id) |
| Current\_stock | INT | NOT NULL |
| Stock\_change\_Id | INT | Ref. Stock\_change\_type (Stock\_change\_type\_id) |
| Stock\_change | INT | NOT NULL |
| Last\_update\_user\_Id | UUID | NOT NULL |
| Last\_update\_dtm | DATETIME | NOT NULL |

1. Shipping: Stores shipment metadata and basic details. Tracking status fetched dynamically via API calls.

| Table: Shipping (Field | Data Type | Constraint) | | |
| --- | --- | --- |
| Shipping\_Id | UUID | Primary Key |
| Inventory\_Id | UUID | Ref. Inventory (Inventory\_Id) |
| Supplier\_Id | UUID | Ref. Supplier (Supplier\_Id) |
| Shipping\_Agency\_Id | UUID | Ref. Shipping\_Agency (Agency\_Id) |
| Tracking\_number | VARCHAR | UNIQUE , NOT NULL |
| Tracking\_Url | VARCHAR | NOT NULL |
| Expected\_delivery\_date | DATE | NOT NULL |
| Last\_API\_Update | DATETIME | NOT NULL |

1. Shipping\_Agency: Stores details of API-based shipping providers.

| Table: Shipping\_Agency (Field | Data Type | Constraint) | | |
| --- | --- | --- |
| Agency\_Id | UUID | Primary Key  Ref. Organization (Organization Id) |
| Agency\_name | VARCHAR | NOT NULL |
| API\_Endpoint | VARCHAR | NOT NULL |
| API\_Key | VARCHAR | NOT NULL |
| API\_Auth\_Type\_Id | INT | NOT NULL  Ref. API\_Auth\_Type (Api\_Auth\_Type\_Id) |

1. Shipping\_Tracking\_Cache: **Temporary storage** for fetched tracking data. This helps to avoid repeated API calls for the same tracking number and provide **temporary fallback** in case of **API failure**.

| Table: Shipping\_Tracking\_Cache (Field | Data Type | Constraint) | | |
| --- | --- | --- |
| Cache\_Id | UUID | Primary Key |
| Shipping\_Id | UUID | Ref. Shipping (Shipping\_Id) |
| Tracking\_number | VARCHAR | NOT NULL, UNIQUE |
| Shipping\_Status | INT | Ref. Shipping\_Status (Shipping\_Status\_Id) |
| Last\_updated | TIMESTAMP | NOT NULL |
| Estimated\_Delivery\_Date | DATE | NOT NULL |
| Location | VARCHAR | NULLABLE |

### Lookup Tables:

1. Supplier\_Compliance : (Compliance\_Type\_Id: INT, Compliance\_Type\_Name: VARCHAR)
   1. Values: Approved, Rejected, Pending, etc.
2. Shipping\_Status : (Shipping\_Status\_Id: INT, Shipping\_Status\_Name: VARCHAR)
   1. Values: Shipped, In Transit, Delivered, etc.
3. Stock\_Change\_Type: (Stock\_Change\_Type\_Id: INT, Stock\_Change\_Type\_Name: VARCHAR)
   1. Values: Restock, Usage, Transfer, etc.
4. API\_Auth\_Type: (Api\_Auth\_Type\_Id: INT, Api\_Auth\_Type\_Name: VARCHAR)
   1. Values: OAuth, Basic, API\_Key

## Donation, Allocation & Fundraising Tables:

### Master Data Tables:

1. Fundraising\_Campaign: Stores fundraising campaigns.

| Table: Fundraising\_campaign (Field | Data Type | Constraint) | | |
| --- | --- | --- |
| Campaign\_Id | UUID | Primary Key  Ref. Organization (Organization Id) |
| Campaign\_name | VARCHAR | NOT NULL |
| Organization\_Id | UUID | Ref. Organization (Organization\_Id) |
| Start\_date | DATE | NOT NULL |
| End\_date | DATE | NOT NULL |
| Target\_Amount | DECIMAL (12, 2) | NOT NULL |
| Raised\_Amount | DECIMAL (12, 2) | NOT NULL |
| Description | VARCHAR |  |

1. Donation: Stores donation transactions.

| Table: Donation (Field | Data Type | Constraint) | | |
| --- | --- | --- |
| Donation\_Id | UUID | Primary Key |
| Donor\_Id | UUID | Ref. Organization (Organization\_Id) |
| Campaign\_Id | UUID | Ref. Fundraising\_campaign (campaign\_Id) |
| Amount | DECIMAL (12, 2) | NOT NULL |
| Donation\_Type\_Id | INT | Ref. Donation\_Type (Donation\_Type\_Id) |
| Payment\_method\_Id | INT | Ref. Payment\_Method (Payment\_method\_Id) |
| Transaction\_date | TIMESTAMP | NOT NULL |
| Receipt\_number | VARCHAR | UNIQUE, NOT NULL |

1. Donation\_Allocation: Tracks fund allocations for donations.

| Table: Donation\_Allocation (Field | Data Type | Constraint) | | |
| --- | --- | --- |
| Allocation\_Id | UUID | Primary Key |
| Donation\_Id | UUID | Ref. Donation (Donation\_Id) |
| Allocation\_category\_Id | INT | NOT NULL |
| Amount | DECIMAL (12,2) | NOT NULL |
| Last\_update\_user\_Id | UUID | NOT NULL |
| Last\_update\_dtm | DATETIME | NOT NULL |

### Lookup Tables:

1. Donation\_Type: (Donation\_Type\_Id: INT, Donation\_Type\_Name: VARCHAR)
   1. Values: One-Time, Recurring, In-Kind, etc.

2. Payment\_Method : (Payment\_Method\_Id: INT, Payment\_Method\_Name: VARCHAR)

1. Values: Wire, Credit Card, etc.

3. Allocation\_Category: (Allocation\_Category\_Id: INT, Allocation\_Category\_Name: VARCHAR)

1. Values: Medical Supplier, Education, etc.

## Historical Tables:

### Master Data Tables

1. Healthcare\_Provider\_History: Tracks which clinics/ hospitals a healthcare provider has worked at over time.

| Table: Healthcare\_Provider\_History(Field | Data Type | Constraint) | | |
| --- | --- | --- |
| History\_Id | UUID | PRIMARY KEY |
| User\_Id | UUID | REFERENCES User\_Profile(User\_Id) |
| Clinic\_location\_Id | UUID | REFERENCES Organization(Organization\_Id) |
| Start\_Date | DATE | NOT NULL |
| End\_Date | DATE | NULLABLE (NULL if currently active) |
| last\_update\_dtm | TIMESTAMP | DEFAULT NOW() |

1. Patient\_Camp\_Visit\_History: Maintains a history of camps attended by a patient.

| Table: Patient\_Camp\_Visit\_History(Field | Data Type | Constraint) | | |
| --- | --- | --- |
| Visit\_Id | UUID | PRIMARY KEY |
| User\_Id (Patient) | UUID | REFERENCES User\_Profile(User\_Id) |
| Camp\_Id | UUID | REFERENCES Camp(Camp\_Id) |
| Visit\_Date | DATE | NOT NULL |
| last\_update\_dtm | TIMESTAMP | DEFAULT NOW() |

1. User\_Organization\_History: Tracks user role changes in organizations. (e.g. : a volunteer moves to caregiver).

| Table: User\_Organization\_History(Field | Data Type | Constraint) | | |
| --- | --- | --- |
| History\_Id | UUID | PRIMARY KEY |
| User\_Id | UUID | REFERENCES User\_Profile(User\_Id) |
| Organization\_Id | UUID | REFERENCES Organization(Organization\_Id) |
| Old\_Role\_Id | INT | REFERENCES User\_Role(Role\_Id) |
| New\_Role\_Id | INT | REFERENCES User\_Role(Role\_Id) |
| Change\_Date | DATE | DEFAULT NOW() |

1. Patient\_Diagnosis\_History: Stores historical patient diagnoses.

| Table: Patient\_Diagnosis\_History(Field | Data Type | Constraint) | | |
| --- | --- | --- |
| History\_Id | UUID | PRIMARY KEY |
| User\_Id (Patient) | UUID | REFERENCES User\_Profile(User\_Id) |
| Test\_Id | UUID | REFERENCES Vision\_Test(Test\_Id) |
| Parameter\_Id | INT | Ref. Test\_Parameter (Parameter\_Id) |
| Parameter\_value | VARCHAR | NOT NULL |
| Diagnosis\_Date | DATE | NOT NULL |
| Diagnosis\_Note | VARCHAR | NULLABLE |
| last\_update\_dtm | TIMESTAMP | DEFAULT NOW() |

1. Patient\_Prescription\_History: Stores past prescriptions for reference.

| Table: Patient\_Prescription\_History(Field | Data Type | Constraint) | | |
| --- | --- | --- |
| History\_Id | UUID | PRIMARY KEY |
| User\_Id (Patient) | UUID | REFERENCES User\_Profile(User\_Id) |
| Prescription\_Id | UUID | REFERENCES Prescription(Prescription\_Id) |
| Prescription\_Date | DATE | NOT NULL |
| Prescription\_Data | JSON | NOT NULL |
| Expiry\_Date | DATE | NULLABLE |
| last\_update\_dtm | TIMESTAMP | DEFAULT NOW() |

1. Inventory\_Stock\_History: Track Stock movement, purchase and consumption over time.

| Table: Inventory\_Stock\_History(Field | Data Type | Constraint) | | |
| --- | --- | --- |
| History\_Id | UUID | PRIMARY KEY |
| Inventory\_Id | UUID | REFERENCES Inventory(Inventory\_Id) |
| Change\_Type | ENUM(Restocked, Used, Adjusted) | NOT NULL |
| Change\_Quantity | INT | NOT NULL |
| Change\_Date | TIMESTAMP | DEFAULT NOW() |
| last\_update\_dtm | TIMESTAMP | DEFAULT NOW() |

1. Supplier\_Compliance\_History: Tracks changes in supplier compliance approvals over time.

| Table: Supplier\_Compliance\_History(Field | Data Type | Constraint) | | |
| --- | --- | --- |
| History\_Id | UUID | PRIMARY KEY |
| Supplier\_Id | UUID | REFERENCES Supplier(Supplier\_Id) |
| Document\_Type | VARCHAR | NOT NULL |
| Status | ENUM(Approved, Pending, Rejected) | DEFAULT 'Pending' |
| Change\_Date | TIMESTAMP | DEFAULT NOW() |

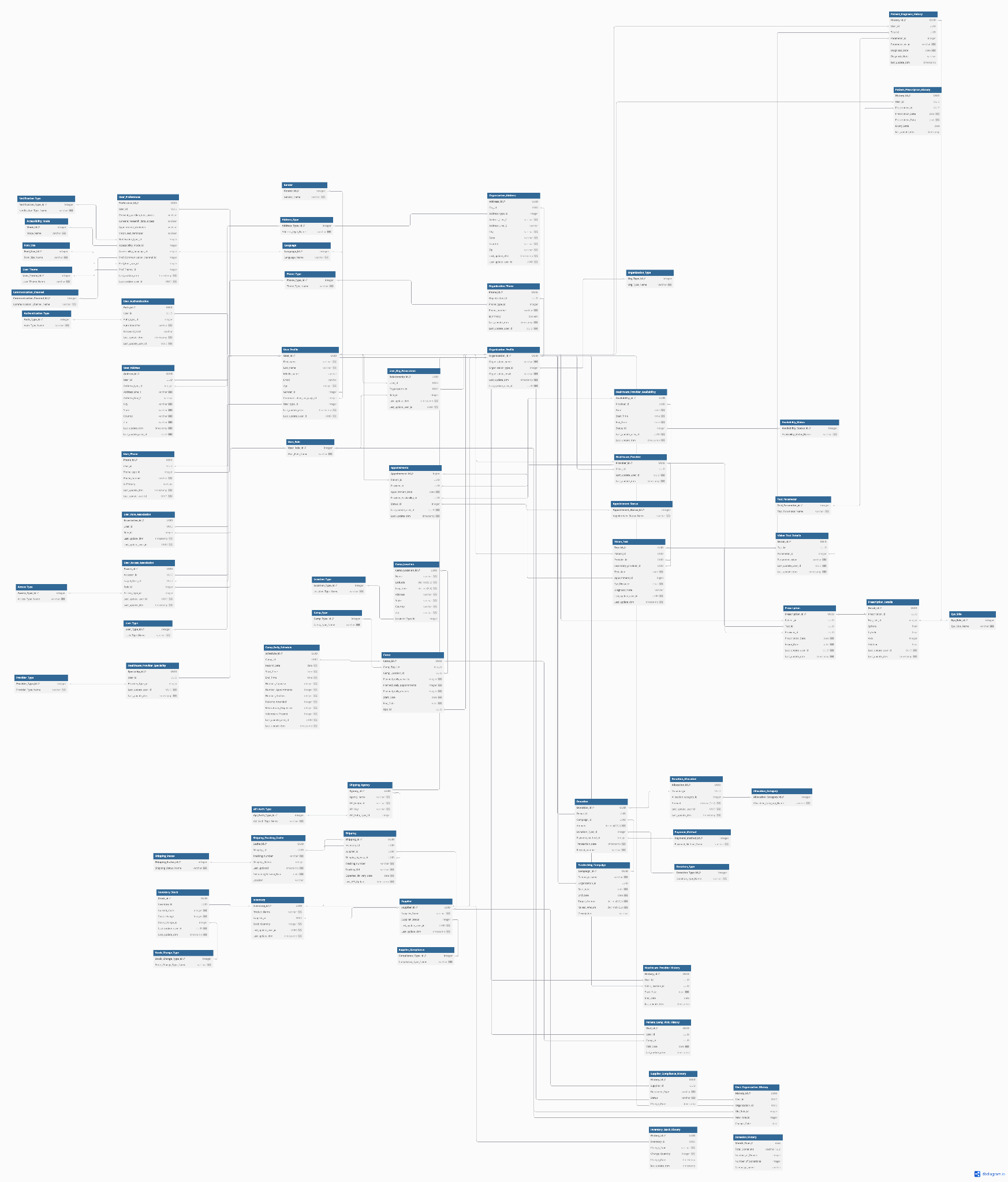
1. Donation\_History: Tracks donations from organizations to iCare.

| Table: Donation\_History | | |
| --- | --- | --- |
| Month\_Year | DATE | PRIMARY KEY |
| Total\_Donations | DECIMAL (12,2) |  |
| Number\_of\_Donors | INT |  |
|  |  |  |
| Number\_of\_Donation | INT |  |
| Campaign\_name | VARCHAR |  |

## 8.1 Data Entities to User Stories map

|  |  |  |
| --- | --- | --- |
| User Stories | Data Entities | Description |
| Vision Screening & Quick Diagnosis | Vision\_Test, Vision\_Test\_Details, Camp, Camp\_Location, Camp\_Daily\_Schedule, User\_Profile | Stores vision test results, screening history, and camp locations for rural users. |
| Appointment Scheduling & Optometrist Interaction | Appointments, Healthcare\_Provider, Healthcare\_Provider\_Availability, Notification\_Type | Tracks optometrist availability, appointments, and automated reminders. |
| Prescription Management & Glasses Ordering | Prescription, Prescription\_Details, Inventory, Shipping | Stores prescriptions, tracks eyewear orders, and manages stock levels. |
| Multilingual & Inclusive Access | User\_Preferences, Language | Enables real-time language translation and multilingual support. |
| Mobile eye checkup camps | Camp, Camp\_Location, Camp\_Daily\_Schedule | Manages camp locations, schedules, and patient participation. |
| Online vision test using images/shapes | Vision\_Test, Vision\_Test\_Details, Prescription | Supports image-based assessments for non-text-dependent screening. |
| AI-driven live translation for communication | User\_Preferences, Language | Ensures smooth communication across languages. |
| Health & Lifestyle Recommendations | Healthcare\_Awareness | Healthcare awareness and educational data. |
| Digital prescription storage | Prescription | Saves patient prescriptions for easy future access. |
| Order tracking for eyeglasses | Shipping | Monitors eyewear shipping and delivery status. |
| Multilingual & Inclusive Access | User\_Preferences, Language | Enables text-to-speech support, local language content, and accessibility features. |
| Audio-based guidance for vision tests | User\_Preferences | Provides voice-assisted test instructions for visually impaired users. |
| Preventive Vision Screening & Monitoring | Vision\_Test, Vision\_Test\_Details | Tracks user eye health changes over time. |
| AI-driven risk factor insights | Vision\_Test, Vision\_Test\_Details | Provides predictive insights into potential vision problems. |
| Vision Health Tracking & Reports | Vision\_Test, Vision\_Test\_Details | Allows users to track vision trends. |
| Manufacturing & Supply Chain Partners | Inventory, Supplier, Shipping, Shipping\_Agency | Manages eyewear production, stock, and distribution. |
| Order management & Inventory | Inventory, Inventory\_Stock | Predicts demand and automates purchase orders. |
| Quality Control & Compliance | Supplier\_Compliance\_History | Ensures manufacturing compliance and product quality standards. |
| Chain of custody tracking | Shipping | Logs production and delivery history for traceability. |
| Sustainability Reporting | Supplier\_Compliance\_History | Tracks eco-friendly material usage. |
| School Vision Programs | Organization\_Profile, Vision\_Test, User\_Org\_association, User\_access\_association, Healthcare\_Awareness | Manages student screenings and parental permissions. |
| Non-Profit Organizations (NPOs) | Organization\_Profile, Donation, Donation\_Allocation | Tracks outreach programs, funding, and beneficiaries. |
| Community Awareness & Engagement | Organization\_Profile | Supports outreach programs and gathers community feedback. |
| Government & Policy Makers | Organization\_Profile, Funding | Provides real-time URE data for decision-making and funding allocation. |
| National URE tracking | Organization\_Profile | Monitors vision health trends across regions. |
| Fundraising & Financial Sustainability | Donation, Fundraising\_Campaign, Donation\_Allocation | Manages donations, sponsorships, and financial planning. |
| Funding allocation | Donation\_Allocation | Optimizes financial resource distribution. |
| Compliance & Data Security | Supplier\_Compliance\_History | Ensures data protection, GDPR/HIPAA compliance, and regulatory reporting. |
| iCare Platform Operations | Organization\_Profile | Monitors app performance, engagement, and operational efficiency. |

## 8.2 Data Architecture Diagrams and description



## 8.3 Analytical Reporting and Architecture

### Analytical Reporting and Architecture

#### Analytical Reporting Needs

The **iCare Vision Platform** requires various analytical reports to track user engagement, healthcare efficiency, supply chain optimization, and financial sustainability. The following reporting areas have been identified:

1. **Vision Test Analytics**
   * **Data Entities**: Vision\_Test, Vision\_Test\_Details, User\_Profile
   * **Metrics**: Number of completed vision tests, diagnosis trends, test completion rates, common vision conditions detected.
2. **Appointment & Consultation Analytics**
   * **Data Entities**: Appointments, Healthcare\_Provider\_Availability, User\_Profile
   * **Metrics**: Total appointments booked, no-show rates, consultation times, provider workload distribution.
3. **Optometrist & Clinic Performance**
   * **Data Entities**: Healthcare\_Provider, Appointments, Prescription
   * **Metrics**: Number of patients seen per provider, diagnosis-to-treatment conversion rates, prescription fulfillment rates.
4. **Community & Camp Engagement**
   * **Data Entities**: Camp, Camp\_Location, Patient\_Camp\_Visit\_History
   * **Metrics**: Number of camps held, attendance rates, follow-up visits, outreach effectiveness.
5. **Supply Chain & Inventory Monitoring**
   * **Data Entities**: Inventory, Supplier, Shipping
   * **Metrics**: Stock levels, shipment tracking, supplier performance, product demand forecasting.
6. **Financial & Fundraising Analytics**
   * **Data Entities**: Donations, Fundraising\_Campaign, Donation\_History
   * **Metrics**: Total funds raised, donor engagement, fund utilization efficiency.

Table 1: Mapping Analytical Reports to Data Entities

|  |  |  |  |
| --- | --- | --- | --- |
| Report Name | Purpose | Data Entities Used | Metrics Tracked |
| Total Vision Tests Taken | Tracks total self-vision and clinic tests | Vision\_Test, User\_Profile | Number of tests, test type (self/clinic) |
| Optometrist Workload Analysis | Tracks optometrist consultations & efficiency | Appointments, Healthcare\_Provider | Total appointments per optometrist |
| Patient Visit Summary | Monitors patient checkups & referrals | Appointments, Patient\_History | Total visits, follow-ups, referrals |
| Test Completion Rate | Measures how many users complete vision tests | Vision\_Test, User\_Profile | % completion rate |
| Top Detected Conditions | Identifies common vision impairments | Vision\_Test, Diagnosis | % of users with Myopia, Astigmatism |
| Conversion Rate to Appointments | Tracks how many users book appointments | Appointments, Vision\_Test | % of test users who booked optometrist |
| Supply Chain & Order Monitoring | Tracks eyeglasses orders & stock levels | Orders, Inventory, Suppliers | Order status, stock availability |
| Donations & Funding Overview | Tracks contributions & funding sources | Donations, Campaigns | Amount donated, top donors, fund usage |

#### Information Architecture for Reporting

The **iCare Analytical Reporting System** is designed with the following structure:

1. **Data Collection**
   * Real-time transactional data from user interactions (appointments, test results).
   * Batch processing of historical data (vision trends, fundraising impact).
   * Integration with external data sources (government health reports, supplier APIs).
2. **Data Storage & Processing**
   * **Operational Database**: **5-NF Tables** for real-time data (Appointments, Prescriptions).
   * **Analytical Data Store**: **Denormalized Tables** for historical data (Donation\_History, Vision\_Test\_History).
   * **Data Warehouse**: Aggregated data for long-term trend analysis.
3. **Data Access & Reporting Layer**
   * **Precomputed Reports**: Scheduled queries to populate dashboards.
   * **On-Demand Reports**: Ad-hoc analysis using SQL-based queries.
   * **AI-Driven Insights**: Machine learning-based prediction for patient risks, supply chain optimization.

Table 2: Information Architecture for Reporting  
This table describes how data flows from collection to reporting.

|  |  |  |
| --- | --- | --- |
| Step | Description | Examples |
| Data Collection | Collects user interactions, vision test results, appointments, etc. | Vision test submissions, appointment bookings |
| Data Storage | Stores structured data in relational DB and logs in NoSQL for performance | Vision\_Test, Appointments, Orders |
| Data Processing | Aggregates raw data into analytical models for reporting | AI models for condition prediction |
| Data Access | Generates dashboards, reports, and real-time analytics | Optometrist workload, patient history |

## Risks

## Risks with Mitigation Strategies

### Technical Risks

**1. Vision Test Accuracy and Standardization**

* **Risk**: The AI-powered vision tests might lack accuracy compared to professional examinations, leading to incorrect diagnoses.
* **Mitigation**:
  + Implement continuous validation against professional examinations
  + Establish minimum device requirements for test validity
  + Create calibration procedures for different devices
  + Regular algorithm updates based on professional feedback

**2. User Data Protection**

* **Risk**: Handling sensitive health data creates privacy and security vulnerabilities.
* **Mitigation**:
  + Implement end-to-end encryption for all health data
  + Adopt a privacy-by-design approach
  + Regular security audits and penetration testing
  + Clear consent mechanisms for data usage
  + Compliance with multiple regulatory frameworks (GDPR, HIPAA)
  + Data minimization practices

**3. System Reliability in Low-Connectivity Areas**

* **Risk**: Service disruptions in rural areas with poor internet connectivity.
* **Mitigation**:
  + Robust offline functionality for core features
  + Data synchronization when connectivity returns
  + Low-bandwidth operation modes
  + Local caching of essential information
  + SMS-based alternatives for critical notifications

### Financial Risks

**4. Financial Sustainability**

* **Risk**: Inadequate funding to maintain operations and scale the platform.
* **Mitigation**:
  + Diversified funding sources (donations, grants, partnerships)
  + Tiered service model with premium features for paying users
  + Corporate sponsorship programs
  + Impact investment attraction
  + Clear metrics demonstrating cost-effectiveness
  + Transparent reporting on fund utilization

**5. Resource Allocation Efficiency**

* **Risk**: Inefficient resource distribution limiting impact and sustainability.
* **Mitigation**:
  + AI-driven demand forecasting for resource planning
  + Data-based prioritization of high-need areas
  + Cost-efficiency metrics for all operations
  + Regular financial audits
  + Lean operational practices

### Operational Risks

**6. Partnership Management**

* **Risk**: Difficulty recruiting and retaining healthcare providers, staff, and volunteers.
* **Mitigation**:
  + Value proposition clearly articulated for each stakeholder
  + Simplified onboarding processes
  + Recognition and incentive programs
  + Continuous engagement strategies
  + Clear performance metrics and feedback
  + Professional development opportunities

**7. Supply Chain and Logistics**

* **Risk**: Delays or failures in eyeglasses production and delivery.
* **Mitigation**:
  + Multiple supplier relationships
  + Inventory management system with safety stocks
  + Quality control processes
  + Distributed production capabilities
  + Alternative delivery methods
  + Clear communication on delivery timelines

**8. Scaling Challenges**

* **Risk**: Inability to scale to meet the needs of 100M potential users.
* **Mitigation**:
  + Modular architecture enabling independent scaling
  + Phased rollout strategy
  + Performance monitoring and optimization
  + Regional deployment strategy
  + Capacity planning based on utilization metrics

### Social and Adoption Risks

**9. Cultural Barriers to Adoption**

* **Risk**: Cultural or social resistance to seeking vision care or using technology.
* **Mitigation**:
  + Culturally sensitive communication strategies
  + Community engagement programs
  + Local champions and success stories
  + Education on vision health importance
  + Family-centered approach to care

## Risks with Limited Solutions

**1. Device Variability**

* **Risk**: The wide variety of mobile devices with different camera and screen specifications affects test accuracy.
* **Partial Mitigation**:
  + Device compatibility testing
  + Calibration procedures
  + Clear indications when a device is unsuitable
  + However, complete standardization across all devices remains challenging

**2. Dependency on Donations**

* **Risk**: Reliance on donations creates unpredictable funding patterns, especially for offline operations.
* **Partial Mitigation**:
  + Diversified funding
  + Operational reserves
  + However, economic downturns can still significantly impact donation levels

**3. Health Awareness Limitations**

* **Risk**: Low awareness about the importance of vision testing, particularly in underserved communities.
* **Partial Mitigation**:
  + Educational campaigns
  + School partnerships
  + However, changing deeply ingrained health behaviors requires long-term cultural shifts beyond the platform's direct control

**4. AI Algorithm Accuracy and Reliability**

* **Risk:** The wide variety of mobile devices with different camera and screen specifications affects test accuracy.
* **Partial Mitigation:**
  + Device compatibility testing
  + Calibration procedures
  + Clear indications when a device is unsuitable

## System level Architectural choices

The iCare system follows a **Microservices Architecture** to ensure scalability, maintainability, and efficient resource management. It also incorporates elements of **Publish-Subscribe/ Pub-Sub Architecture** to support real-time notifications and data synchronization across multiple components, such as appointment scheduling and inventory updates.

### Technology-Related Options

* **Cloud-Based Solution:** iCare leverages a cloud-based infrastructure for scalability, resilience, and remote accessibility. This enables seamless data storage, efficient processing, and integration with external APIs.
* **Mobile & Web Accessibility:** The system supports both web and mobile applications to ensure accessibility across different user demographics, particularly in underserved areas.
* **Event-Driven Design:** The system leverages event-driven architecture allowing seamless communication between modules like inventory management, order processing, and real-time notifications.

### Security Considerations

* **Data Encryption:** All sensitive data, including user information and medical records, is encrypted both at rest and in transit to maintain confidentiality.
* **Role-Based Access Control (RBAC):** The system enforces fine-grained access control to ensure that users, healthcare providers, and administrators only have access to relevant data.
* **Compliance with Regulations:** Although the application initially targets India, our design aligns with privacy regulations such as **GDPR** and **HIPAA**, ensuring patient data protection and secure data-sharing policies.

### Usability Considerations

* **Multilingual & Inclusive Design:** The system supports multiple native indian languages and accessibility features, such as text-to-speech and AI-driven translations, ensuring a seamless experience for diverse users.
* **Intuitive UI/UX:** A user-friendly interface with guided workflows simplifies processes like booking appointments, tracking prescriptions, and managing donations.
* **Offline Mode Support:** In areas with limited connectivity, the mobile app includes offline capabilities, allowing users to store and sync data when reconnected.

### Additional System-Wide Considerations

* **Scalability:** The cloud-based microservices architecture ensures that the platform can handle an increasing number of users, partners, and transactions efficiently.
* **Interoperability:** The system is designed to integrate with third-party healthcare solutions, government databases, and NGO platforms to facilitate seamless data exchange.
* **AI-Driven Insights:** AI capabilities are embedded for predictive analytics, demand forecasting, and risk assessment in vision care management.

This structured approach ensures that iCare’s system architecture aligns with both functional and non-functional requirements, delivering a secure, scalable, and user-centric experience.

## Applicable Standards

* **HIPAA (Health Insurance Portability and Accountability Act)**: Requirements for protecting sensitive patient health information, including Privacy Rule and Security Rule compliance.
* **GDPR (General Data Protection Regulation)**: Requirements for data protection and privacy for individuals within the European Union.

## Definitions, Acronyms, Initialisms, and Abbreviations (DAIAs)

The table/link below contains a collection of Definitions, Acronyms, Initialisms and Abbreviations specific to this project.

|  |  |  |
| --- | --- | --- |
| **DAIAs** | **Term** | **Definition** |
| e.g DTC | Direct to Consumer | The product is shipped directly to the customer instead of being picked from the store |

## 

## References

Kleppmann, M. (2017). Designing data-intensive applications: The big ideas behind reliable, scalable, and maintainable systems. O'Reilly Media.

ADDENDUM – Some Quality Attribute Definitions

These are some quality attributes (QA) with definitions and an example. You must pick the right attributes for your solution.

**Portability**

How many other platforms is this system expected to run on? What browsers and devices is it compatible with?

Example of Portability: The system is expected to be deployable in many geographies. The system needs to support internationalization. Localization of content is to be supported.

Device Portability - The system must be accessible from multiple mobile operating systems such as Windows, IOS and Android.

**Performance**

How does the system perform under a particular workload? How many concurrent users does it support? How many transactions per second would this system support?  What are system requirements for data volume, throughput and response time?

Example:

 The Response time for any click on all the web pages should be no more than 0.5 seconds.

The system should be able to support a maximum of 300 concurrent users.

The system database should handle 3000 update requests (workload performance).

**Usability**

Ease with which people can achieve a particular task/goal using the application. Can users complete the intended task at their intended speed? How much training does the user need?

Example:

Any functionality from the main page should be reached in maximum 3 user clicks.

Any delay in retrieving data extending more than 1 sec should pop up a window “Please Wait..”

All screens should have a help button.

**Testability**

Degree of effectiveness and efficiency with which test criteria can be established for a system, product or component and tests can be performed to determine whether those criteria have been met.

Example:

All user stories must be written such that test cases can be created to cover 90% of the workflows.

The development plan must ensure that test cases for each component must be verifiable as it is developed.

**Scalability**

Scalability is the capability of a system, network, or process to handle a growing amount of work, or its potential to be enlarged to accommodate that growth.

Example:

The system must support 10,000 users and scale to 1.5 times its initial size in the period of 1 year.

The system must support data growth  to 2X its initial size  in the second year.

System should be able to scale out horizontally to support load during peak time.

Agility

Agility in supporting new business models, enables service discovery and provides visibility into business process health.

Example:

System should support e-commerce order integration after a business expansion project which is due in 6 months.

It must be possible to incorporate state-based tax laws in 15 days as the system is deployed in various states.

Supportability (aka Maintainability)

How effectively a software system or component can be kept running after deployment, based on resources that include quality documentation, diagnostic information, and both knowledgeable and available technical staff.

Example:

The system should have a built-in mechanism to send alerts when the pressure reading exceeds 200 psi.

Adequate documentation should be provided for conducting diagnostic analysis and Root Cause Analysis in case of an error.

The system should maintain persistent logs of every functional operation.

Flexibility

The ability for the solution to adapt to possible or future changes in its requirements.

Example:

The System should be flexible to align with the new requirement by the state of California for audits which is due in June 2021.

Sustainability

Architecting systems for sustainability minimizes use of resources used and can result in more cost-effective solutions.

Example:

The system workload should be optimized for demand to increase energy efficiency and minimize the total resources deployed.

Example:

Use software architecture that supports data access and storage requirements to minimize use of compute, network and storage resources.