

## CareDraw:

*Smart Tech. Safe Care. Delivered to You.*

### Contents

SECTION 1 – SCOPE AND SCENARIO .....	2
1.1 - About.....	2
1.2 - Caregiving & Home Health Fact Sheet: .....	2
SECTION 2 - OVERVIEW.....	3
2.1 – Mission Statement .....	3
SECTION 3 – USER STORIES.....	3
3.1 – Phlebotomist .....	3
3.2 – Caregiver .....	3
SECTION 4 – FUNCTIONAL REQUIREMENTS.....	3
4.1 - Customer Portal.....	3
4.2 - Phlebotomist Workforce Platform .....	4
4.3 - CareDraw Operations Management .....	4
SECTION 5 – USE CASES.....	4
5.1 - Booking.....	4
5.2 - Chatbot.....	5
5.3 – Sequence Diagram.....	6
SECTION 6 – CLASS MODEL.....	6
6.1 – Activity Diagram.....	7
6.2 – Focused Sequence Diagram.....	9
6.3 – Usable Code Sample.....	9
SECTION 7 – MODEL DISCUSSION.....	11
7.1.1 – Specimen Design Goals .....	11
7.1.2 – Specimen Design Tradeoffs .....	11
7.2.1 – User Design Goals.....	11

7.2.2 – User Design Tradeoffs .....	12
7.3.1 – Patient Design Goals .....	12
7.3.2 – Patient Design Tradeoffs.....	12
7.4.1 – Billing Design Goals.....	12
7.4.2 – Billing Design Tradeoffs .....	12
7.5.1 – Support Design Goals.....	13
7.5.2 – Support Design Tradeoffs .....	13
SECTION 8 – STATE TRANSITION DIAGRAM .....	13
8.1 – CHATBOT SUB-STATES.....	14
SECTION 9 – GUI SKETCH .....	14
SECTION 10 – NON-FUNCTIONAL REQUIREMENTS .....	15
10.1 - HIPAA Compliance .....	15
10.2 – Low Latency Notifications .....	15
REFERENCES .....	15

## SECTION 1 – SCOPE AND SCENARIO

### 1.1 - About

CareDraw is a scheduling and operations platform that connects 1099 phlebotomists with patients needing in-home lab draws – with a particular focus on homebound seniors and individuals living with chronic illness. CareDraw was founded by caregivers determined to make healthcare more accessible, personal and convenient; we understand the challenges that something as simple as bloodwork can pose and have made it our mission to help alleviate those and ease the burden of caregiving.

### 1.2 - Caregiving & Home Health Fact Sheet:

- By 2040, the number of Americans aged 65 or older is expected to reach 84 million – increasing from 56 million in 2020 [1].
- Patients who leverage for home healthcare services following hospital discharge experience lower rates of readmission, fewer ED visits, lower costs and higher satisfaction of care [1].
- The US home care testing market was valued at over \$10B in 2023 and is anticipated to grow at an annual rate of almost 7% over the next 10 years [2].
- Currently, over 50 million adults function as unpaid caregivers, providing regular assistance with daily living activities and personal care [3].

- Over 40% of caregivers report low overall well-being, with caregivers having higher rates of anxiety, depression, substance abuse and leaves-of-absence from work [4].

## SECTION 2 - OVERVIEW

### 2.1 – Mission Statement

The mission of CareDraw is to make in-home blood draws simple, reliable, and dignified for patients who are homebound, chronically ill, or aging in place. By connecting a trusted network of phlebotomists with families through our scheduling and operations platform, CareDraw brings clinical-quality lab services directly to the home. We integrate IoT-enabled tools to ensure blood sample integrity, while generative AI streamlines documentation, FAQs and communication for patients, providers, and caregivers. In doing so, we aim to reduce caregiver burnout, improve continuity of care, and help patients stay healthy where they feel the safest—at home.

## SECTION 3 – USER STORIES

### 3.1 – Phlebotomist

As a phlebotomist, I want a simple and convenient way to pick-up extra work without the commitment of a full clinic-based shift, so that I may use my training to help people in need while earning additional income.

### 3.2 – Caregiver

As the primary caregiver for my aging mother and full-time employee, I want a platform that allows me to schedule in-home bloodwork, so I can ensure my mother gets the care she needs without having to coordinate transportation or take off work.

## SECTION 4 – FUNCTIONAL REQUIREMENTS

### 4.1 - Customer Portal

1. **Manage Profile:** Users will be able to create profiles containing addresses, contact information, clinical needs, insurance and billing information, and relevant details.
2. **Schedule Bloodwork:** Users will be able to view available appointment slots, set preferred dates/times, review fasting requirements, note any special needs, and cancel/reschedule appointments.
3. **View Lab Results:** Users will be able to view Lab Results from past bloodwork appointments.
4. **Chat with CareDraw AI Chatbot:** Users will be able to engage with GenAI-powered chatbot to ask questions and receive assistance on platform functionality, CareDraw services or care-related educational resources.

5. **Message CareDraw Team:** Users will be able to message CareDraw Support staff with issues or questions that cannot be addressed via chatbot – this may include positive/negative reviews about Phlebotomists, complex billing questions or specific requests.

## 4.2 - Phlebotomist Workforce Platform

6. **Manage Profile:** Phlebotomists will be able to create profiles including clinical credentialing, contact information and set preferred work hours or routes.
7. **Job Acceptance/Dispatch:** Phlebotomists will be able to review open requests for blood draws, locations, requirements and accept/decline jobs.
8. **Job History:** Phlebotomists will be able to review completed job history and visit notes for reconciliation against earnings or future job planning/forecasting.
9. **Earnings & Tax:** Phlebotomists can view paychecks and obtain 1099 Tax documents for reporting.

## 4.3 - CareDraw Operations Management

10. **IoT Blood Coolers:** CareDraw Operations Managers can review IoT Smart Cooler details such as temperature, tampering detection and GPS details to ensure blood sample integrity and compliant chain-of-custody.
11. **Message Users:** CareDraw Support Staff can field and respond to customer inquiries.
12. **Platform Administration:** CareDraw Administrators can view all bloodwork requests, completed jobs, customer and phlebotomist profiles, and lab results.
13. **Reporting/Dashboarding:** CareDraw Leadership will be able to view financial and operational reporting and dashboarding as it relates to visits, staff, customers and insurance claims.

# SECTION 5 – USE CASES

## 5.1 - Booking

<b>Use case Name</b>	Caregiver Appointment Booking and Results Review	
<b>Actor:</b>	Caregiver	
<b>Description:</b>	This use case describes the event of a patient's primary caregiver booking an appointment, engaging in conversation with the CareDraw chatbot, reviewing lab results and submitting payment for services rendered	
<b>Pre-condition:</b>	Caregiver is logged into the CareDraw platform	
<b>Step #</b>	<b>Actor</b>	<b>System</b>
1	Caregiver selects 'Schedule Appointment' module from CareDraw homepage	Launches scheduling GUI and presents Caregiver with calendar containing available time slots during which Phlebotomist may arrive, as well as lab order options.
2	Caregiver selects preferred timeslot and labs to be completed from prefixed list (CBC, Metabolic, A1C, lipid, etc.)	Confirms appointment window and lab order, solicits caregiver for special needs and provides fasting instructions
3	In free-form text box, Caregiver enters any special needs or requests and confirms appointment	Provides confirmation of appointment window to patient, generates open request in Phlebotomist job dispatch dashboard.
4		Upon Phlebotomist acceptance, sends appointment confirmation to Caregiver and Patient, removes open request from job dispatch dashboard
5		System retrieves insurance information from patient profile to begin claim/bill process

6		Upon Appointment conclusion, sends post-visit summary to Caregiver and Patient, indicating Results are pending
7		System constructs claim/bill with services rendered – routes to insurance or direct to Caregiver/Patient Billing Module in event of 'cash-pay'.
8		System pushes Lab Results to Patient record, notifies Caregiver and Patient
9	Caregiver clicks on 'chat bubble' icon launching GenAI-powered Chatbot	System displays chat window and greets user with a welcome prompt
10	Caregiver solicits for patients most recent A1C reading and what it means	System queries Lab Results database table for most recent A1C level and provides brief note about whether reading is elevated, normal or low.
11	Caregiver asks how reading compares with historical levels	System queries Lab Results table for most recent 5 readings and displays in simple tabular format for readability. System also provides link and brief instruction on how Caregiver can access lab results in platform.
12	Caregiver inquires as to whether they should be concerned.	System indicates lab results can be influenced by a variety of factors and recommends reaching out to care team for additional consultation.
13	Caregiver selects provided link and is navigated to Lab Results module, reviews full results from recent visit.	System saves chat conversation to patient account for future reference by Caregiver/Patient
14	Caregiver navigates to 'Billing' module, selects bill pertaining to recent visit	System returns itemized detail of charges owed for services rendered – noting what patient owes vs. what insurance has paid (if applicable)
15	Caregiver selects 'Pay Now' and submits payment	System processes payment, saves payment confirmation to Patient transaction history in Billing module
<b>Alternate Courses:</b>	<p>Alt Step #1 (System/Caregiver) – If System identifies Caregiver as being linked to more than 1 patient, Caregiver is presented with GUI window to select appropriate patient for appointment</p> <p>Alt Step #12 (System) – If system interprets levels as being in 'normal' range, response shall include reassurance that readings fall within normal range, but consultation with care team is still advised.</p>	

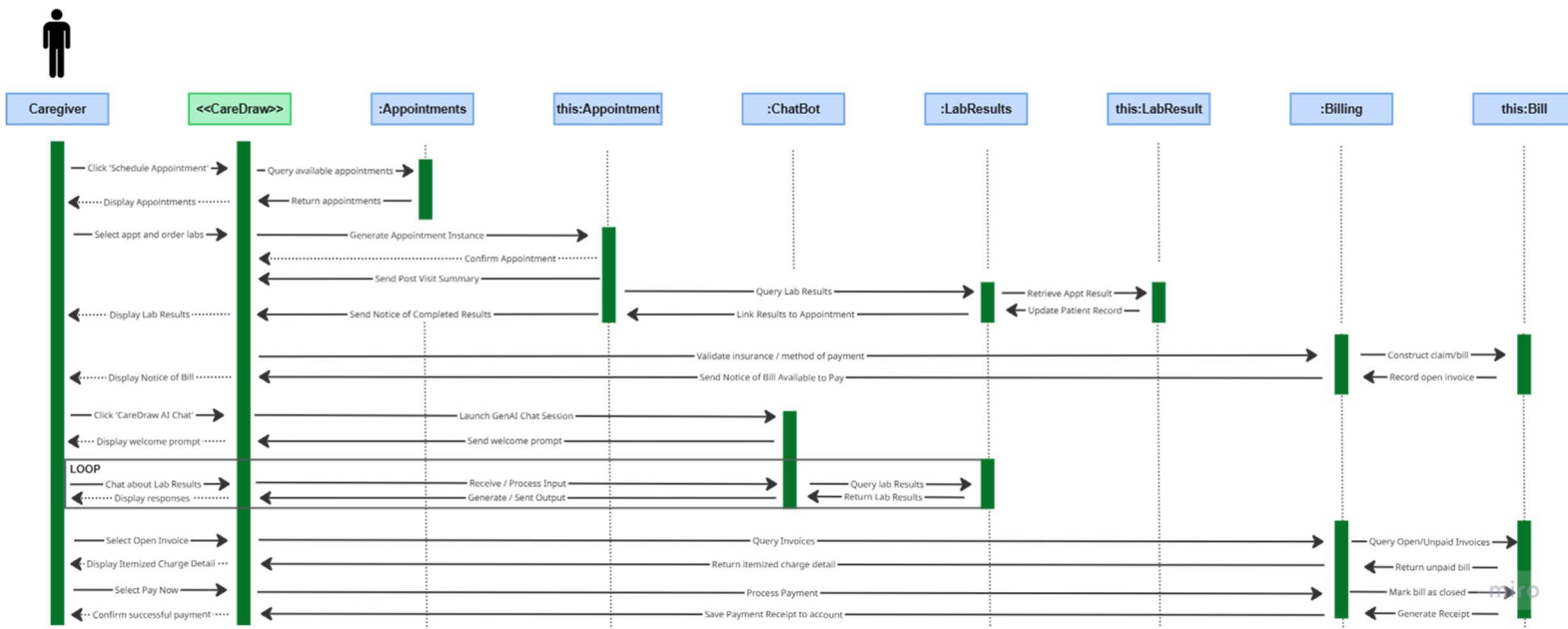
## 5.2 - Chatbot

<b>Use case Name</b>	Caregiver engages with CareDraw GenAI Chatbot	
<b>Actor:</b>	Caregiver	
<b>Description:</b>	This use case describes the event of a patient's primary caregiver initiating a conversation with the CareDraw chatbot, referencing recent lab work and soliciting help with interpreting the results.	
<b>Pre-condition:</b>	Caregiver is logged into the CareDraw platform	
<b>Step #</b>	<b>Actor</b>	<b>System</b>
1	Caregiver clicks on 'chat bubble' icon launching GenAI-powered Chatbot	System displays chat window and greets user with a welcome prompt
2	Caregiver solicits for patients most recent A1C reading and what it means	System queries Lab Results database table for most recent A1C level and provides brief note about whether reading is elevated, normal or low.
3	Caregiver asks how reading compares with historical levels	System queries Lab Results table for most recent 5 readings and displays in simple tabular format for readability. System also provides link and brief instruction on how Caregiver can access lab results in platform.

4	Caregiver inquires as to whether they should be concerned.	System indicates lab results can be influenced by a variety of factors and recommends reaching out to care team for additional consultation.
5	Caregiver indicates end of conversation.	System saves conversation to patient account for future reference by Caregiver/Patient
<b>Alternate Courses:</b>	<p>Alt Step #4 (System) – If system interprets levels as being in 'normal' range, response shall include reassurance that readings fall within normal range, but consultation with care team is still advised.</p> <p>Alt Step #5 (Actor/System) – Patient closes chat window without indicating end of conversation; system saves conversation to patient account upon termination of chat session.</p>	

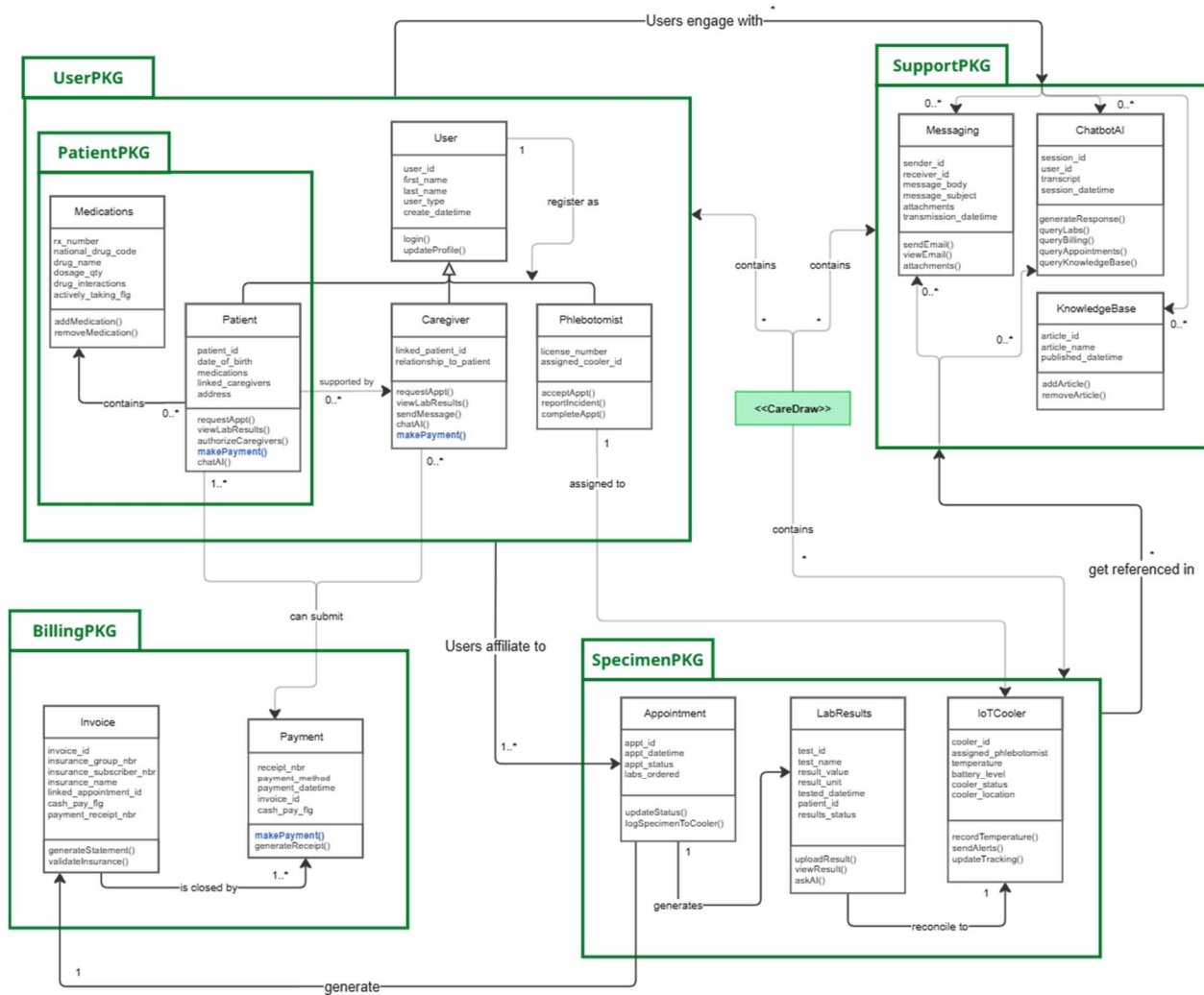
## 5.3 – Sequence Diagram

The sequence diagram pictured below provides additional context to the Use Case described in Section 5.1 and captures a common workflow within the CareDraw platform. Here we follow the end-to-end sequence -- from scheduling to payment -- for a caregiver acting on behalf of a patient they are authorized with. While of course not every sequence will follow this same pattern of actions, the depiction here highlights several of the platform's key objects and illustrates how the supporting functions drive outcomes and actions while in-platform – understanding these entities allows us to extrapolate to other sequences and use cases.



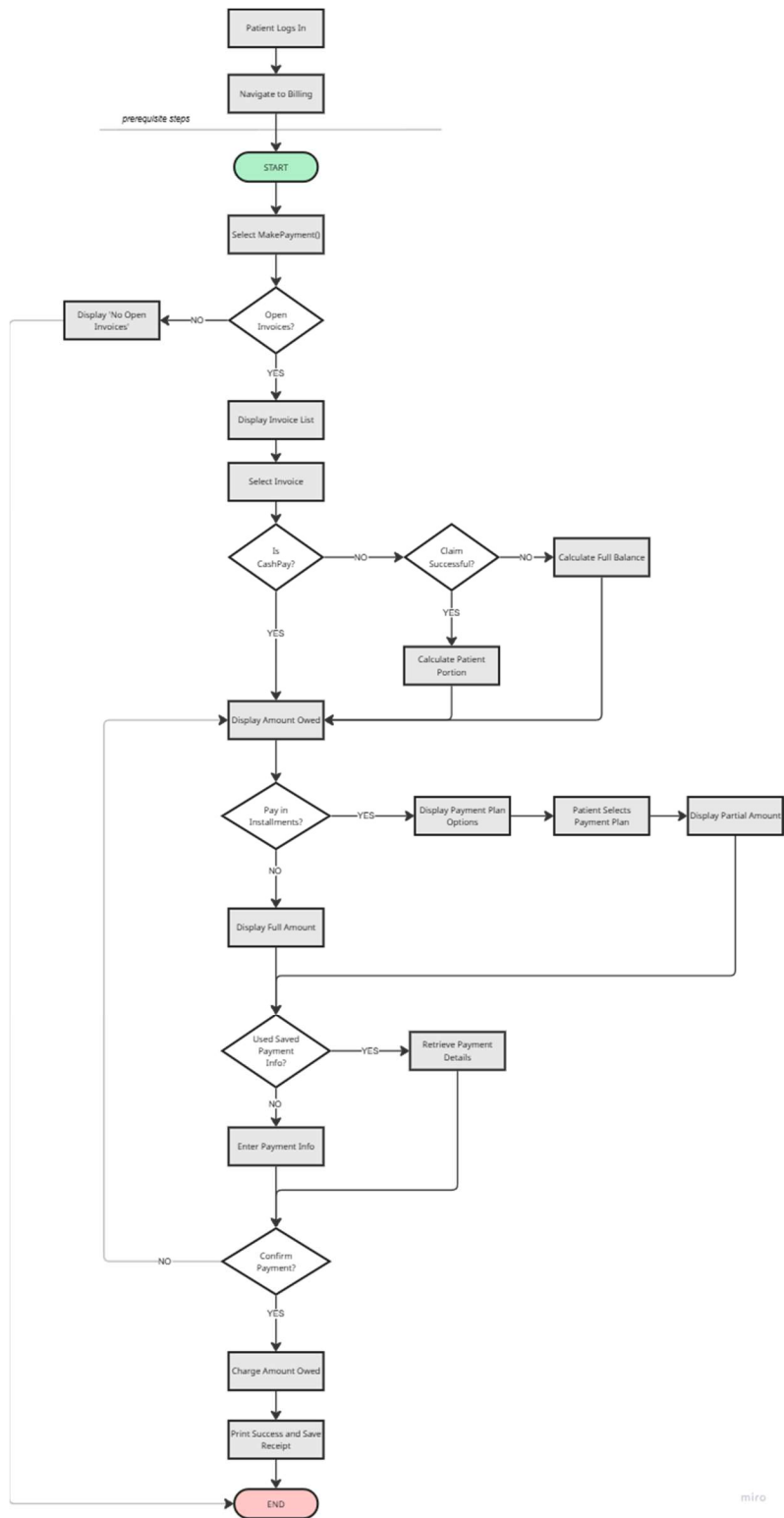
## SECTION 6 – CLASS MODEL

The diagram below depicts the core Classes that constitute the structure of the CareDraw platform, along with several of the key packages, attributes and methods. This model is further examined and contextualized in the subsequent sections detail below.



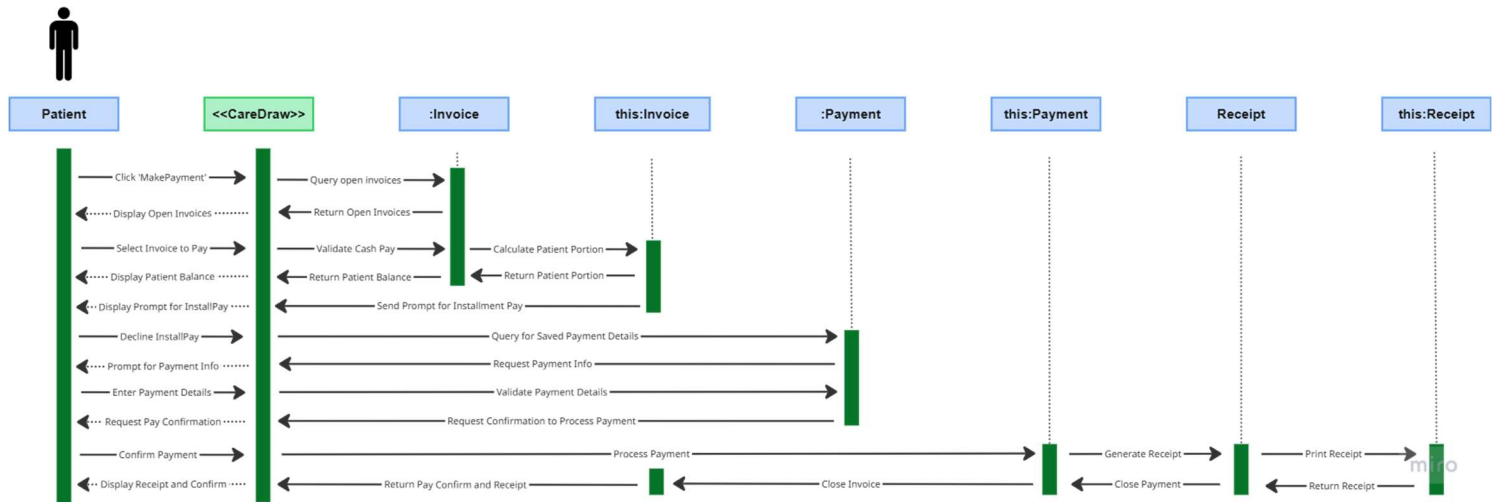
## 6.1 – Activity Diagram

Here we review the `makePayment()` method, which enables Patients and Caregivers to submit payment for services rendered. To illustrate, the activity diagram below follows a patient who is logged into the CareDraw platform, has navigated to the Billing module and has invoked or executed this function by selecting the Make Payment button - note the 'prerequisite steps' are present for context only and are not bundled in the `makePayment()` function.



## 6.2 – Focused Sequence Diagram

To further exemplify the `makePayment()` method, consider the below diagram, which depicts the Activity Diagram in focused, sequential form. By examining the method in this manner, we can differentiate and understand the key actions from both the patient (or client) side and the backend (or server) side, in linear form. The CareDraw design UI is represented here in green, with patient actions showing to the left and system actions to the right.



## 6.3 – Usable Code Sample

The python function pictured below adapts the `makePayment()` method into a usable piece of code. This function takes in a list of unpaid invoices, as well as parameters such as selected invoice, installment choice, confirmation status, and optional saved payment information. The function then applies business rules, such as insurance validation, to return a structured response with status, messages and receipt details. In current state, the function is not built to involve any interactivity, though that could certainly be refactored in. This is intended to be a prototype only.

```

from typing import List, Dict, Optional
from pprint import pprint

def make_payment(
    invoices: List[Dict],
    selected_invoice_id: int,
    use_installments: bool = False,
    confirm: bool = True,
    saved_payment_info: Optional[Dict] = None,
) -> Dict:

    # Check if invoices exists
    if not invoices:
        return {"status": "no_invoices", "message": "You have no open invoices."}

    # Validate invoice selection
    selected_invoice = next((inv for inv in invoices if inv["id"] == selected_invoice_id), None)
    if not selected_invoice:
        return {"status": "error", "message": "Invalid invoice selection."}

    # Insurance / Cash pay validation
    if selected_invoice["cash_pay_flg"] == 0:
        if selected_invoice.get("claimApproved") == 1:
            balance_due = selected_invoice["amount"] * 0.2 # assume 80% coverage
            msg = "Insurance processed. Adjusted balance applied."
        else:
            balance_due = selected_invoice["amount"]
            msg = "Insurance denied. Full payment required."
    else:
        balance_due = selected_invoice["amount"]
        msg = "Cash pay invoice. Full payment required."

    # Install or full pay
    if use_installments:
        amount_to_charge = balance_due / 3 # sample with 3 installments
        plan_info = {"installments": 3, "per_payment": amount_to_charge}
    else:
        amount_to_charge = balance_due
        plan_info = None

    payment_info = saved_payment_info if saved_payment_info else {"method": "manual_entry_required"}

    if not confirm:
        return {
            "status": "cancelled",
            "message": "Payment not confirmed. Returning to payment options.",
            "invoice_id": selected_invoice_id
        }

    # Process payment + receipt
    receipt = {
        "invoice_id": selected_invoice["id"],
        "amount_paid": amount_to_charge,
        "plan_info": plan_info,
        "payment_method": payment_info,
    }

    return {
        "status": "success",
        "message": "Payment successful.",
        "invoice_id": selected_invoice["id"],
        "receipt": receipt
    }

# invoke fn:
if __name__ == "__main__":
    invoices = [
        {"id": 1, "amount": 200, "cash_pay_flg": 0, "claimApproved": 1},
        {"id": 2, "amount": 150, "cash_pay_flg": 1, "claimApproved": None},
    ]

    result = make_payment(invoices, selected_invoice_id=1, use_installments=True, confirm=True)
    pprint(result)

```

```

{'invoice_id': 1,
 'message': 'Payment successful.',
 'receipt': {'amount_paid': 13.333333333333334,
             'invoice_id': 1,
             'payment_method': {'method': 'manual_entry_required'},
             'plan_info': {'installments': 3,
                           'per_payment': 13.333333333333334},
             'status': 'success'}

```

## SECTION 7 – MODEL DISCUSSION

### 7.1.1 – Specimen Design Goals

The Specimen Package brings together core classes—Appointment, LabResult, and IoT Cooler—that collectively manage the full, end-to-end lifecycle of laboratory specimens in a distributed care model. Each class is purposefully designed to work seamlessly with the others, ensuring that responsibilities are clearly defined and cohesive with real-world workflows. This tight internal alignment enhances the clarity and maintainability of the package, while well-defined interfaces limit the dependencies on external packages, making the system more adaptable to future changes. The inclusion of IoT-enabled cooler tracking further extends the package’s ability to scale and support future monitoring without disrupting core scheduling or results processing logic. Through its structure, the package promotes modular development, simplifies testing, and supports broader reusability. By organizing related functionality into a focused and self-contained unit, the design reinforces both clarity and adaptability of the application’s specimen-handling capabilities.

### 7.1.2 – Specimen Design Tradeoffs

Association with Users gives Appointment the benefit of seamless coordination and role-based actions but introduces the risk of becoming overly dependent on this class - overtaxing this may impact efficiency.

Delineation between Lab Results and other system classes sufficiently partitions sensitive and protected health data, but reference from the Support package raises Security concerns.

Modularizing IoT Cooler establishes a pathway for future IoT product support, such as wearables, but carries a higher barrier to implementation due to complexity and may exhibit unreliability in early stages.

### 7.2.1 – User Design Goals

The User package serves as the foundational layer for managing system access, identity, and role-based behaviors across the CareDraw platform. By centralizing the User superclass, shared attributes such as authentication credentials, contact information, and permissions can be uniformly handled, reducing redundancy. In turn, the subclass roles of Patient, Caregiver and Phlebotomist inherit from this superclass, allowing for tailored behavior without compromising cohesion. This modular separation supports flexibility in expanding user types in the future, while maintaining low coupling with other system packages. The design promotes clarity in role responsibilities, streamlining both access control and interaction tracking within the broader application architecture.

## 7.2.2 – User Design Tradeoffs

Centralizing shared User attributes promotes code reusability, but downstream inheritance may limit scalability in the future if role types expand and require more complex or nuanced data modeling.

### 7.3.1 – Patient Design Goals

As a subpackage of User, Patient isolates domain-specific functionality related to those receiving care. It contains the Patient class, which manages health-related attributes and histories, as well as the Medications class, which handles both current prescriptions and historically administered treatments. This separation ensures that health context remains decoupled from general user behavior, fostering a clean and maintainable design. By organizing patient-specific logic into its own subpackage, the system achieves high cohesion, making it easier to implement features like medication tracking, care plan updates, or clinical alerts without impacting other user types. The structure anticipates future scalability while preserving a clear boundary between administrative and clinical responsibilities.

### 7.3.2 – Patient Design Tradeoffs

Isolating Patient creates opportunities for personalization and exceptional user experience but may become overly coupled with clinical classes given associations with Medications, LabResults and Appointments.

### 7.4.1 – Billing Design Goals

The Billing package is designed to encapsulate financial transactions in a modular and consistent manner, isolating payment logic from clinical or user-facing workflows. The Invoice class draws upon Appointment data to automatically generate structured billing records, promoting consistency and reducing redundancy across the system. Tight cohesion within the package ensures that financial responsibilities — invoice creation, balance tracking, and payment status — are managed in a unified context. Meanwhile, loose coupling with external packages, like Appointment or User, allows billing rules to evolve independently from clinical workflows or user roles. This separation also supports extensibility, such as integrating third-party payment processors or advanced insurance logic in future iterations. Overall, this package reinforces modularity, reusability, and financial accountability without entangling core healthcare operations.

### 7.4.2 – Billing Design Tradeoffs

Invoice class helps separate billing from clinical operations, but critical outside dependency on Appointment increases coupling.

Payment class simplifies transactions allowing for easy reporting and auditing on balances owed and payment history; however, partial payments, reimbursements and failed transactions may increase complexity and impact usability.

## 7.5.1 – Support Design Goals

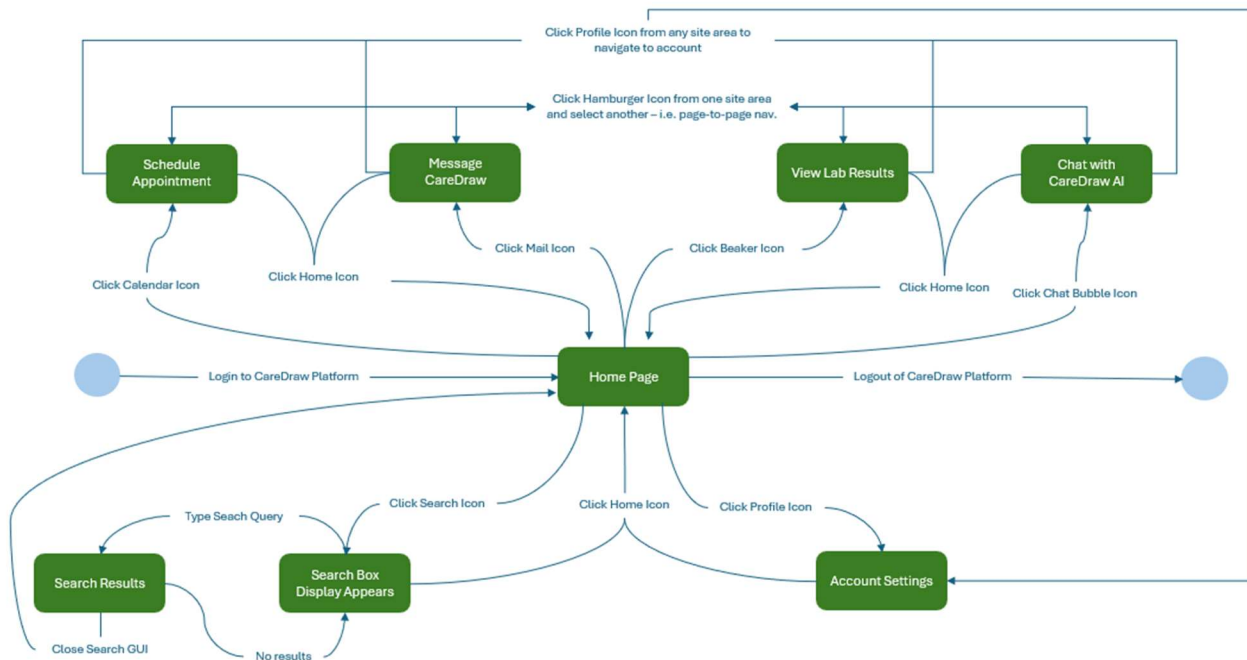
The Support package centralizes auxiliary services that enhance user interaction and system responsiveness. It includes the ChatbotAI, KnowledgeBase, and Messaging classes, which work together to provide intelligent assistance, facilitate communication, and deliver context-aware support. Users can engage with the package through natural language queries or real-time message exchanges, ensuring help is accessible without disrupting primary workflows. The package is intentionally decoupled from domain-specific logic, allowing it to serve as an interoperable layer that can evolve independently or scale across modules. This promotes system adaptability, encourages code reuse, and maintains clean separation between operational functionality and user-facing support.

## 7.5.2 – Support Design Tradeoffs

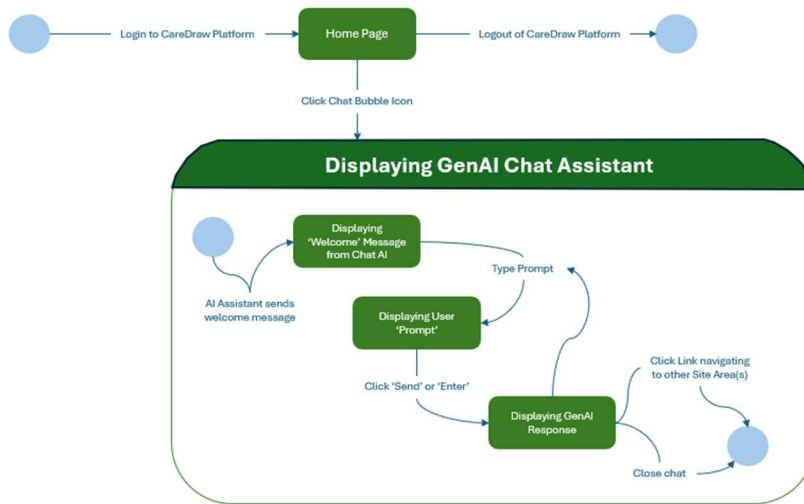
Highly available and reliable support channels promote automation and scalability, but robustness of support is contingent on information available via outside packages – which increases coupling.

Accuracy and effectiveness of Chatbot AI may require significant capital and technological investment, as low investment may yield poorer response output and ultimately low customer experience scores.

## SECTION 8 – STATE TRANSITION DIAGRAM

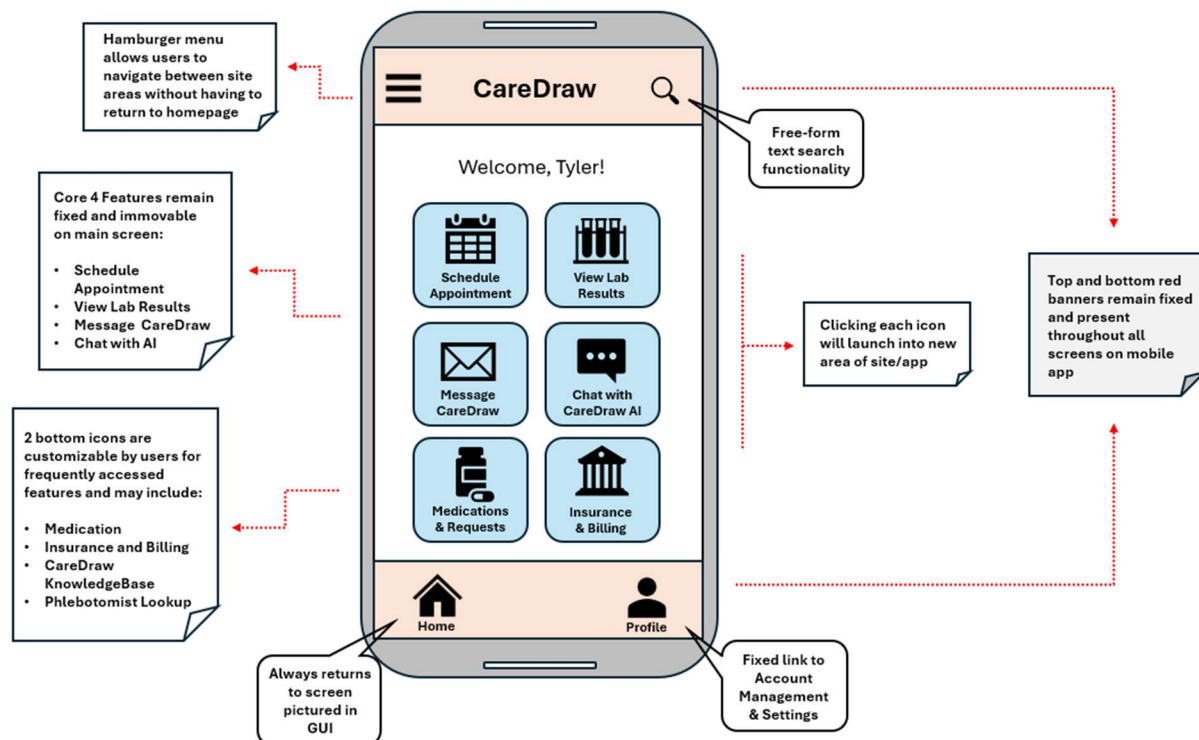


## 8.1 – CHATBOT SUB-STATES



## SECTION 9 – GUI SKETCH

The below GUI displays the Home / Landing Page as it appears for Patients/Caregivers using the Mobile version of CareDraw.



## SECTION 10 – NON-FUNCTIONAL REQUIREMENTS

### 10.1 - HIPAA Compliance

It is essential that all in-home blood draw samples and related data maintain compliance with HIPAA requirements. For CareDraw, compliance in this regard is a non-negotiable given that the platform directly handles PHI via patient data, medical history, specimen collection and lab results. Violating HIPAA can result in federal penalties, destroy customer credibility and trust, and ultimately force the business to cease operations.

### 10.2 – Low Latency Notifications

The CareDraw platform must maintain  $\leq 5$  second transmission times as it relates to appointment updates and visit arrival tracking to ensure real-time coordination between patients, caregivers and phlebotomists.

## REFERENCES:

- [1] GrandViewResearch. (n.d.). *US Home Healthcare Services Market Size, Share & Trends Analysis Report by Type (Skilled, Unskilled), And Segment Forecasts, 2021-2028*. Market Analysis Report. <https://www.grandviewresearch.com/industry-analysis/us-home-healthcare-services-market-report>
- [2] Global Market Insights (July 2024). *Home Care Testing Market - By Product Type (Strip, Cassette, Test Panel, Midstream, Dip Card), Test Type (Infectious Disease, Diabetes & Glucose, HIV Pregnancy, Cholesterol & Triglycerides, UTI), Sample (Blood), Distribution Channel & Forecast, 2024 – 2032*. Report ID: GMI10201. <https://www.gminsights.com/industry-analysis/home-care-testing-market>
- [3] US Centers for Disease Control and Prevention. (3 September 2024). *Dementia Caregiving as a Public Health Strategy*. Public Health. <https://www.cdc.gov/caregiving/php/public-health-strategy/index.html>
- [4] GuardianLife. (n.d.). *Standing Up and Stepping In: A modern look at caregivers in the US*. <https://www.guardianlife.com/reports/caregiving-in-america>