Milestone 1

Domain Modeling and Initial System Design

Team SAAK

Simon Chu, Andy Hammer, Anthony Meza, Kevin Wang

Domain Model

Model:

During the process of creating a domain model, several relevant entities were identified in order to properly represent the system with which the software interacts. In Figure 1, we depict the final domain model created[1]. A notable design choice made is that healthcare workers (such as nurses and doctors), do not interact with the scheduling software directly. Instead, administrators who work at the hospitals will manage the software and work with the healthcare staff to properly schedule patients.

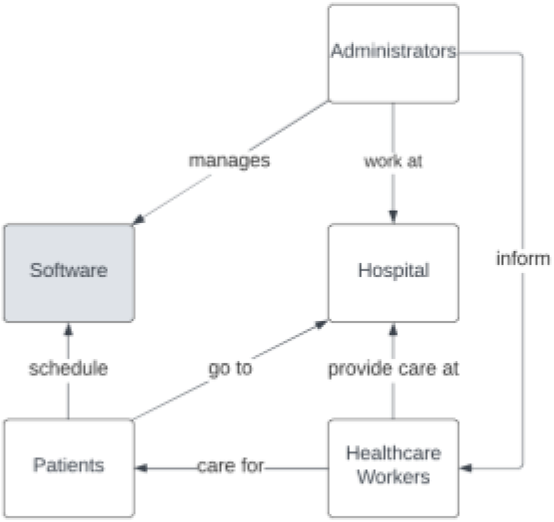

Figure 1: Domain Model of System

System Requirements:

1. Software shall provide patients with all currently available appointments from which to select.
2. Software shall require patients to log into the system before selecting an appointment slot.
3. Software shall ensure selected appointments are associated with the location that patient attends/selects for their medical care needs.
4. Software shall have all appointments associated with the doctor and nurse who provides care to the patient.

Assumptions Regarding the System:

1. Patients have experience with electronic devices, including utilizing web applications.
2. Patients have a reliable internet connection during the time they utilize the scheduling software.
3. Hospitals and healthcare workers operate on the standard working schedule and are available for appointments between nine a.m. to five p.m.
4. The hospitals and healthcare locations have the required equipment, inventory, and time to adequately treat patients.
5. There exists a time slot that works for the patient, and if not, the software will not be used and the patient will contact the administration in another manner.

[1] In a break from convention, we utilize directional connections in the domain model. This allows for easier reading of connection arrows, and should not imply other additional information.

6. All payment and insurance options will be dealt with in person or via another contact method and not require the software to intervene, beyond the potential input of the patient's insurance plan.
7. Administrators will know healthcare workers' schedules and manage their schedules for them.

Alternate Design Decisions

One design decision considered was that some healthcare workers might want to attend to their own schedules and access the software themselves. While most doctors and some nurses have administrative staff to assist with scheduling, it is not beyond reason that they might need to have their login. However, it can be assumed that in that case, the person will be acting as both a healthcare worker as well as a system administrator. Including a relationship between the healthcare workers and the software in the domain, model would increase the reading difficulty of the model as is, and so it was decided not to be included.

Another design alternative was to include electronic health records (EHR) within the domain model. Since the software could require access to EHRs in the future, it is possible to incorporate them into the design model. However, it was decided not to include them, as it will likely not greatly improve understanding of how the software is incorporated into the healthcare system, but could potentially heighten confusion.

Quality Attribute Scenarios

High Priority Attributes

1) The system must be sufficiently reliable for users so that if an appointment is scheduled and confirmed by our application, there is a 99.99% chance that this time slot will be appropriately reserved without issue (Reliability).

- Justification: Our application is being specifically created to handle appointment schedules for patients, which means that ensuring these appointments are scheduled as desired is of the highest priority. Our requirements for our application also describe ensuring we are meeting patient expectations when creating an appointment.

2) The system ensures users' personal information is protected by only allowing information to be visible to authorized users after being given permission (Security).

- Justification: Health information is private and sensitive information that is legally required to be inaccessible by unauthorized users.

3) For users who have an account in our system, the service must be available at all times (24 hours a day, and every day of the week), with the exception of scheduled maintenance. For such maintenance, users must be informed 24 hours in advance, and take place during times of low traffic (Availability).

- Justification: Since our application will be holding people's personal health information, it is much more straightforward for them to be able to access it at any time during the day with little to no restrictions. People also can get sick at any time during the day, so making sure they have the opportunity to schedule appointments or check prior appointments makes our application desirable to use.

Intermediate Priority Attributes

4) Efficiently handles appointment requests from users with a max slowdown of 5 seconds for response time from the application when taking user input. (Performance)

- Justification: It is important to consider response time because if an application takes too long to respond, clients are going to be less inclined to use it. 5 seconds is a satisfying enough metric since more than that would ruin the user experience, but also provide enough flexibility for implementation. This is not the highest priority simply because it is not needed to satisfy functionality, but is needed to make it a desirable application to use.

5) Make the system straightforward and easy to use, ensuring at a minimum a 90% satisfaction rating for everyone who uses our system to manage appointments (Usability).

- Justification: We want to make sure people want to use our application, which means that it needs to at the very least be user-friendly. This is not a high priority though since our end goal is to make sure appointments are booked efficiently and straightforwardly, but we want people to want to use the system, so it also needs some level of usability.

Low Priority Attributes

6) Our system can be extended to other types of testing/vaccination appointments as they are requested by stakeholders with minimal effort on behalf of developers (Extensibility).

- Justification: This is an important requirement to keep in mind when developing software since it is also foreshadowed that we will need to add more components to our software later in the semester, but the other requirements are our current focus for our implementation

7) Our system can promptly respond to appointment cancellations due to emergency schedule changes for any party involved. An alert to those affected should be sent within 8 hours to allow for rescheduling (Flexibility).

- Justification: Cancellations are important to consider, but with our time constraints for developing the application, making sure core functionality is established takes greater priority over emergencies that might influence scheduling.

Component Design

Components

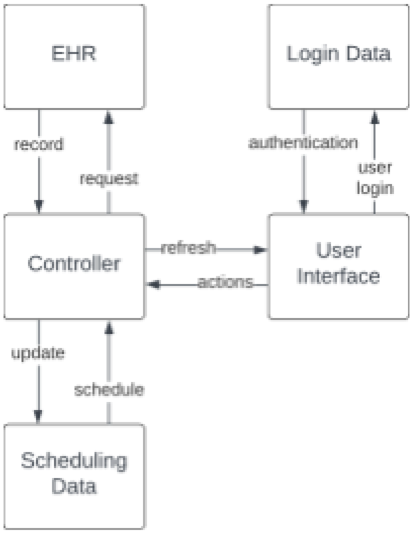

Figure 2: Component Diagram

1. **User Interface:** The user interface is the display that user interacts with, in which they can log in, select their appointment slot, and fill out all of their information.
2. **Controller:** The controller is the central component that processes user actions from the user interface, requests and receives data from EHR, updates scheduling data, and refreshes the user interface with the latest data. It acts as the mediator among different components.
3. **Login Data:** The login data manages user credentials and performs authentication checks when a user tries to log in via the user interface. It ensures that only authorized users can access the system.
4. **Scheduling Data:** The scheduling data manages data related to scheduling, such as appointments. It updates its data based on the updated actions from the controller and maintains the schedule.
5. **Electronic Health Record (EHR):** EHR is responsible for storing and managing patient health records, which healthcare workers in hospital facilities can access through this system. It can be a software system managed by a third party, and it will respond to requests from other systems via protocols like HTTP / HTTPS.

Interfaces

1. **Controller - User interface:** This interface allows the user interface to send user actions to the controller to be processed, and the controller to send updated data back to the user interface to refresh what is displayed.
2. **User interface - Login Data:** This interface is used for authentication purposes. When a user attempts to log in through the user interface, the data is sent to the login data component for authentication. The results of this authentication are then sent back to the user interface.
3. **Controller - Scheduling Data:** This interface allows the controller to update the scheduling data. The update action from the controller sends new or modified data to the scheduling data component, which can be used for the appointment scheduling functionality.
4. **Controller - EHR:** This interface allows the controller to send requests to the EHR for patient records and receive these records.

Alternative Design Decisions

An alternative design decision was a more generic web application architecture including an App interface, a frontend server, a backend server, and a database. This architecture is more generic and could apply to a wide range of web applications. However, it does not indicate a specialized application with components tailored for appointment scheduling or healthcare data management which are important requirements to this project. In summary, our finalized design would be more suitable for a specialized scheduling system in the field of healthcare that requires careful handling of sensitive data, while the alternative design is more flexible and generic, suitable for a broad range of web applications. The decision between these two would largely depend on the specific requirements and domain of the application being developed.

Data Model

User: A person who is able to log in to use the system.

Patient: A user who logs into the system to schedule the appointments.

Administrator: The people who handle scheduling for doctors and nurses.

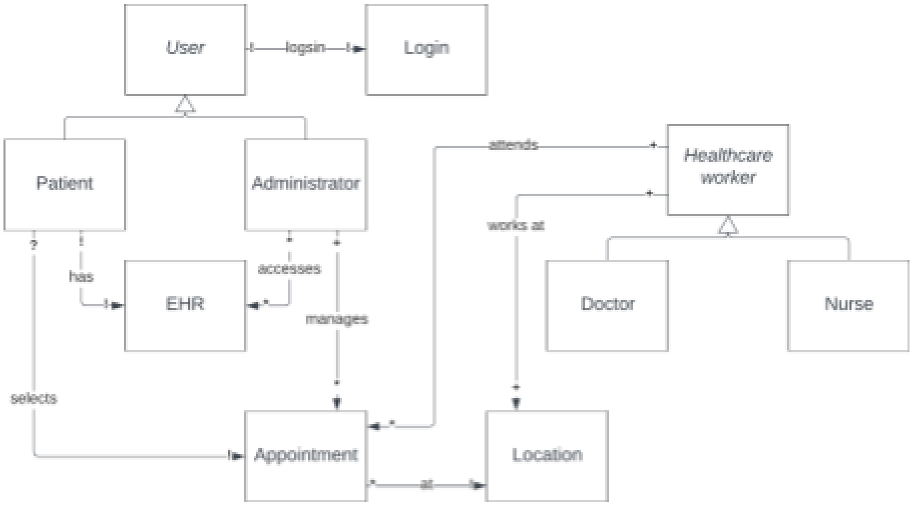

Figure 3: Data Model

EHR: Electronic Health Records, which stores patients' data, including their health records and billing records.

Log-In: Stores the credentials for logging in for both patients and administrators.

Healthcare workers: People who work at the hospital and provide medical care, including doctors and nurses.

Appointment: One-hour long time slots that are agreed upon between the patients and the healthcare worker for them to provide care for the patients.

Location: The location where the appointments are, including the hospital address.

Design Decisions

Decision 1

We design the system so that the administrator decides on a fixed set of appointments and finds the corresponding healthcare workers to be assigned for the appointments, and then released to the patients to sign up. The decision is to ensure that there's a way to coordinate the matching of appointments.

Decision 2

We assume that the healthcare provider does not directly perform scheduling. The administrative staff (administrator) handles all the scheduling-related tasks, assigns corresponding staff, and releases appointments to patients. Therefore, no interface is necessary to be provided to the nurses/doctors to schedule the appointments. All communications between the nurses/doctors and administrative staff are conducted via email communication. This is to reduce the overhead of implementing an additional interface for nurses and doctors (with different permissions), and makes sure each appointment is assigned to the right person/level of staff.

Alternative Designs Decisions

Alternative Decision 1:

One of the alternative design decisions we considered is whether there are at least one/one or more (+) healthcare workers attending the appointment, or if there should be zero or more (*). This is assuming that healthcare workers release the appointments first, and patients then sign up for them.

Alternative Decision 2:

We assume that the administrator handles all scheduling-related matters, including informing the healthcare workers to provide their availability and details of the appointments to the responsible personnel.

Alternatively, we can provide an additional interface, which can retrieve the login information for both nurses and doctors and reduce the cost of hiring administrative staff.