NABS

The Neurosurgery Appointment Booking System

Adrien Lindner
Leah Musie
Melissa Sjostrom
Ryan Conley
Tavish Wille

I. INTRODUCTION

> *The Department of Neurological Surgery at The Ohio State University Wexner Medical Center is led by world-class leaders in academic research, education and medicine. Our multidisciplinary teams pioneer therapies on every neurological front, such as stroke, neuromodulation, endoscopic skull base approaches, spinal cord and brain implants and other complex operations.*
>
> *Ohio State's Neurosurgery residency program, fellowships and education programs provide countless opportunities for collaborative and focused learning at every level — including seminars, lectures and journal clubs. As part of Ohio State's Neurological Institute, we train tomorrow's leaders and advance neurological medicine with a collaborative, cross-disciplinary approach.*
>
> - *OSU Wexner, Department of Neurological Surgery Website*

The Ohio State University Wexner Medical Center (OSUMC) Neurosurgery Department is a world class institute. However, the process by which patients are referred to the department from outside physicians is outdated and overly complicated. In fact, the current system is often ignored because of how cumbersome it is. This team's Capstone project was to demonstrate a proof of concept of a streamlined referral system through the use of a web application. The concept would demonstrate that an automated system could replace the current referral system. Ultimately the team wanted to create a prototype that could be delivered to an external developer to continue development.

Initially, the team was given a overarching concept of how the web application should interact with the user. Over the course of the semester, this was expanded upon and shaped into a proof of concept. The student team worked in conjunction with an external developer for the first half of the project, but largely progressed independently. The streamlined web application was designed to be used by physicians and their clinic staff while arranging meetings with a patient.

A patient's information is entered, and a referral is created based on a physician selected or a particular diagnosis. Initially, this referral would be sent to a referred clinic, in this case being the OSU Wexner Department of Neurosurgery. After the referral is created and sent (placed into "sent" status), it is ready for a physician or staff, at a clinic to which the patient is being referred, to accept or decline the referral. In the case of acceptance, the referred clinic contacts the patient and proceeds to schedule them.

II. REQUIREMENTS

Problem Statement

The OSUMC Neurosurgery Department is a world class institute. However, the process by which patients are referred to the department from outside physicians is outdated and overly complicated. In fact, the current system is often ignored because of how cumbersome it is. This team's Capstone project was to demonstrate a proof of concept of a streamlined referral system through the use of a web application.

Initial Functional Requirements

As originally presented, the desired products of NABS included an iOS and Android app as well as a web app. Physicians and staff needed to be able to create and manage their own accounts, and be able to create a referral request while in the room with the patient. They needed to be able to select a physician to refer to by specialty and schedule availability, and optionally be able to mark the request as “urgent”. A physician’s office that was accepting referrals needed, in addition to the ability to view, accept and deny incoming referrals, to be able to specify what manner of previous testing results or imaging the patient would need to have prior to requesting an appointment, and the referring physician’s office would need to be able to transmit said data through the app. (This indicated that HIPAA compliance would be a major factor in later phases of the project.) Automated scheduling into IHIS (the hospital’s appointment scheduling system) was also desired.

Final Functional Requirements

The scope of NABS had some revisions as the semester progressed. Upon interfacing with Ontoborn (an external company hired to also work on the project), it was decided that with their staff and resources, the iOS and Android app portions of the project were to be assigned to them. Automated scheduling was delayed until a future version of the app, as was transmission of imaging and test results, which potentially included very large files.

Epics and User Stories

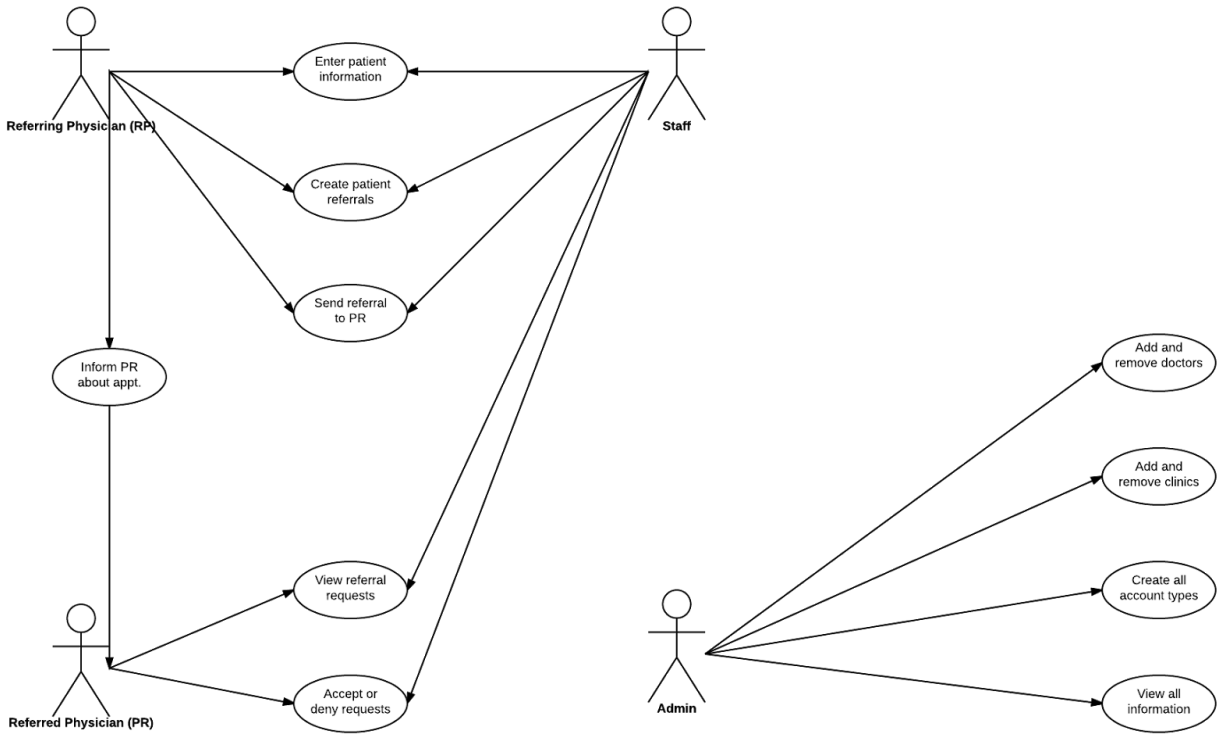
As part of our functional requirements gathering, we collected user story “epics” from our sponsor and worked with Ontoborn to compile a complete list of user stories. The following epics, broken apart into user stories, are samples of ways in which the actors (the different account types; doctors, staff, admins) may interact with and benefit from the system.

- Epic: As a physician receiving referrals[1], I want to schedule patients.
    - User story: As a physician receiving referrals, I want to schedule patients so that I don’t have big gaps in my schedule.
    - User story: As a physician receiving referrals, I want to schedule patients as soon as possible so that I can have higher patient satisfaction.
    - User story: As a physician receiving referrals, I want to schedule patients so I can spend less time playing phone tag and more time with them.
    - User story: As a physician receiving referrals, I want to schedule patients during the time that I am available for clinical duties, not surgical, so that I do not have to reschedule them.
    - User story: As a physician receiving referrals, I want to be able to modify patient requests, and inform them of it, in the case of schedule changes.
- Epic: As a referring physician, I want to be able to refer a patient to the right doctor.
    - User story: As a referring physician, I want to be able to schedule PR by specialty, availability, and specifically so that I can ensure my patient is referred to the best available doctor for their condition.
    - User story: As a referring physician, I want to be able to enter patient health information so that the referred physician is able to better help my patient.
- Epic: As a referring physician, I want to be able to contact the referred physician directly after scheduling my patient.
    - User story: As a referring physician, I want to be able to contact the referred physician so that I can inform them about the patient’s condition and health information.
    - User story: As a referring physician, I want to be able to be aware of the referred physician’s requirements so that I can share the necessary information in time for the appointment.

[1] Note: Staff and Physicians have much of the same duties, so their user stories overlap.

    - User story: As a referring physician, I want to be able to contact the referred physician so that we can straighten out any remaining requirements.
- Epic: As a physician, I want to be able to add staff to act on my behalf in the system.
    - User story: As a physician, I want to be able to add and authorize staff to act on my behalf so that more referrals can be completed, efficiently and quickly.
- Epic: As an admin, I want to be able to maintain the system.
    - User story: As an admin, I want to be able to see all doctors' information (profile) so that I can modify them.
    - User story: As an admin, I want to be able to remove/add doctors so that the scheduling app can remain up to date.
    - User story: As an admin, I want control over accounts so that I can approve/remove/modify them when necessary.

Use Case Diagram



The Use Case Diagram describes the set of actions (use cases) that are available to each of the users of the system (the actors). The use cases shown above are based on our gathered user stories; both serve as a means of describing the ways in which each actor may interact with the system.

Actors and summary of actions:

- Referring physicians - capable of creating and forwarding patient referrals to referred physicians.
- Referred physicians - receives referrals from referring physicians, may accept, modify, or reject them.
- Staff - hospital staff members, who act on the doctor's behalf in creating, accepting, and denying referrals.
- Admins - system administrators, capable of performing all administrative actions, such as creating, modifying, and viewing accounts.

Non-Functional Requirements

NABS is an modern solution to an outdated and time-consuming system. In order to maximize the time savings and error reduction that it promises, NABS needs to be easy and intuitive to use for all users. To reduce time waste, there needs to be a clear and simple sequence of events to follow in order to create a referral. To maintain a pleasant experience, screens need to not be cluttered, and follow a logical pattern using common UX themes and indicators, such as icons to represent common functions. In order to be used across different machine and systems, it also needs to be adaptable and responsive, being able to accommodate both computer and tablet screens.

It also needs to be as accessible and well-documented as possible for maintenance by future administrators. If NABS is a success in the Neurosurgery Department, it may be adapted for use in other departments. Designing NABS to make that expansion and modification easier is desired as well.

Out Of Scope

As mentioned previously, several of the initial goals of NABS were dropped due to collaborative planning, resource availability, and time constraints.

When it appeared that Ontoborn would be working with us for the majority of the semester and continuing work afterward, the iOS and Android app portions of the project were taken on by them, and the backend work was to be split. Therefore, when Ontoborn was removed from the project late in the semester, the iOS and Android apps were dropped completely. Integration with IHIS for scheduling purposes appeared to be too great a challenge without Ontoborn's help, so that was also left for a future version of NABS. Finally, transmission of patient imaging was set aside for the time being due to the expected size of such files.

What was left was largely user experience, referral form field standardization, and referral workflow-related. While slightly unfinished, the sponsor was informed of these changes to scope throughout the project, so expectations were mitigated.

III. PROJECT MANAGEMENT

Methodology

General Agile practices were adopted for the development of NABS. However, due to the interfacing with followed by the separation from Ontoborn, some aspects changed over the course of the semester.

Initially, we set out with one-week sprints in mind, but once we began to collaborate with Ontoborn, we changed to two-week sprints to be in accordance with them. When Ontoborn was no longer involved with NABS, and there were only about four weeks left in the semester, we reordered back into one-week sprints. Similarly, when we began to create plans for sprints and assign tasks, we created a Trello board. At Ontoborn's behest, we switched to Freedcamp; later, we switched back to Trello.

At the outset, we planned to employ the principles of test-driven development for NABS. However, since we did not begin coding until quite late in the semester, and had the unexpected challenge of re-organizing our approach when Ontoborn was removed from the project, test-driven development did not turn out to be a practical approach. Functional and manual acceptance testing was applied to the project at this point and was used until the end of the project.

Resources

Our sponsor was very responsive and encouraging as we gathered requirements from him. His domain knowledge and willingness to meet repeatedly, in person and online, to discuss his vision for NABS was indispensable to our development process.

Out of the five members of our team, four had taken the web application development prerequisite to this course. Therefore, the majority of the team had experience developing a web application specifically with Ruby on Rails. This made the project far more manageable once it had been redefined to exclude the iOS and Android applications from the project scope. All members of the team had prior knowledge with agile team management systems. This experience was invaluable when the scope of the project was changed so abruptly due to Ontoborn's departure.

Our contacts at Ontoborn were excellent resources as well. They met with us online at least once a week, helped to elicit and clarify requirements from the sponsor, and encouraged us to stay organized and on schedule. Even after their funding was cancelled, they reached out to offer any help should we need it.

Finally, our professor and Capstone coordinator was an excellent mentor. He helped us stay focused and reasonable, stepped in to help figure out what was going on when Ontoborn was removed from the project, and provided us with valuable feedback every step of the way.

Iteration Plan and Timeline

August 22nd: Project started; initially tried to proceed individually until it was revealed that an commercial team, Ontoborn, based in India, was in negotiations to take over the project.

September 13th: Interfaced with Ontoborn for the first time, and began collaborating with them; requirements gathering.

October 30th: Informed by Ontoborn that funding was halted, independent sprint/iteration plan created.

Sprint 1 (October 31st - November 5th)

- Finalize database structure
- Sign in view
- Create user controller
- Create staff model
- Implement devise and sign in logic
- Create doctor model
- Profile view
- Develop tests for sprint 2

Sprint 2 (November 6th - November 12th)

- Referral model
- Referral view
- Switch database from SQLite to PostrgreSQL
- Hospital-side view
- Implement sending emails to users
- Implement permissions of users
- Develop tests for sprint 3

Sprint 3 (November 13th - November 19th)

- Implement auto-population of data in form based on account signed in
- Implement pending/in-progress view for referrals
- Implement search/selection view for doctors
- Implement sending emails to patients when referral is accepted or denied

Sprint 4 (November 20th - November 26th)

- Finish writing tests
- Fix remaining bugs
- Hardening sprint

Risk Management

The team primarily used mitigation techniques to manage risk. Everybody was involved in all aspects of the project so that way in the event one person could not continue work due to scheduling or other reasons, it did not put the entire project off track. Further, the team was very deliberate in including everyone in all emails. This way there would be a clear paper trail if there was any conflict over what was assigned to whom. This was

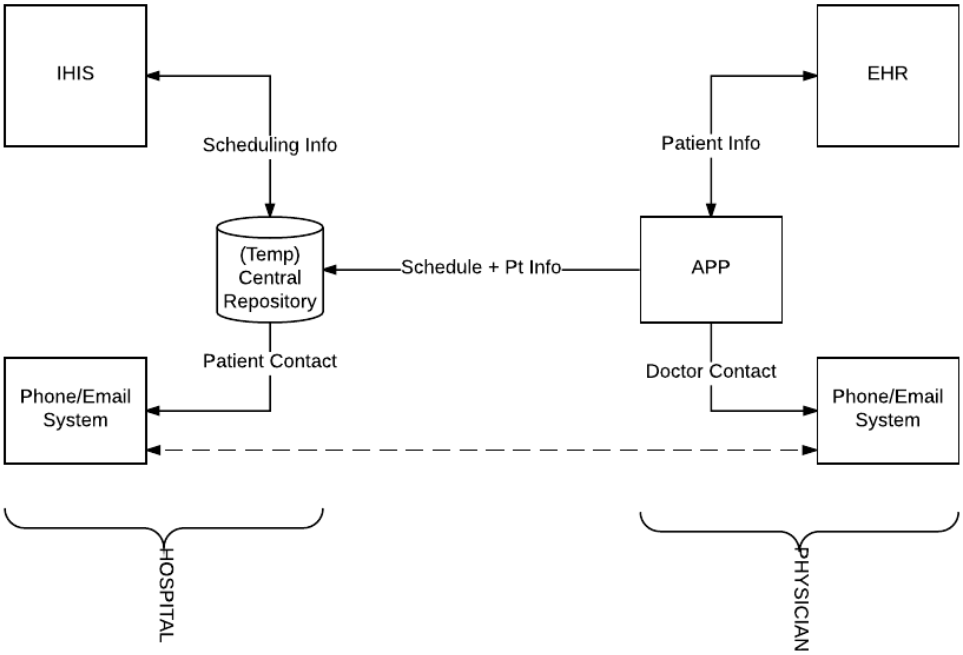
especially important when considering the work that needed to be done by people outside the team. The team made sure to also arrange consistent meeting with the external parties involved to avoid unverified features and correct requirements.

Testing

Before we started coding, we created a test plan that included a test description, expected behavior, and potential severity for each use case. When we first started coding, we would manually check if it has the expected behavior and would only merge it into the repository if it worked correctly. Later on, we wrote automated tests using RSpec to test the models and controller methods.

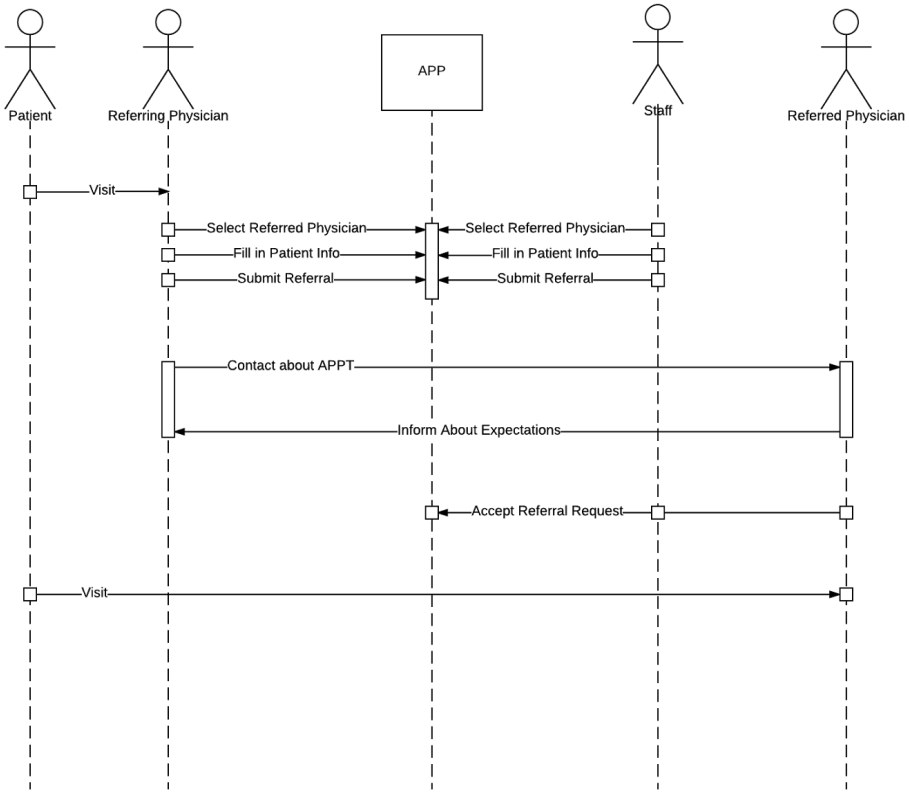
IV. ARCHITECTURE AND DESIGN

System Architecture Diagram



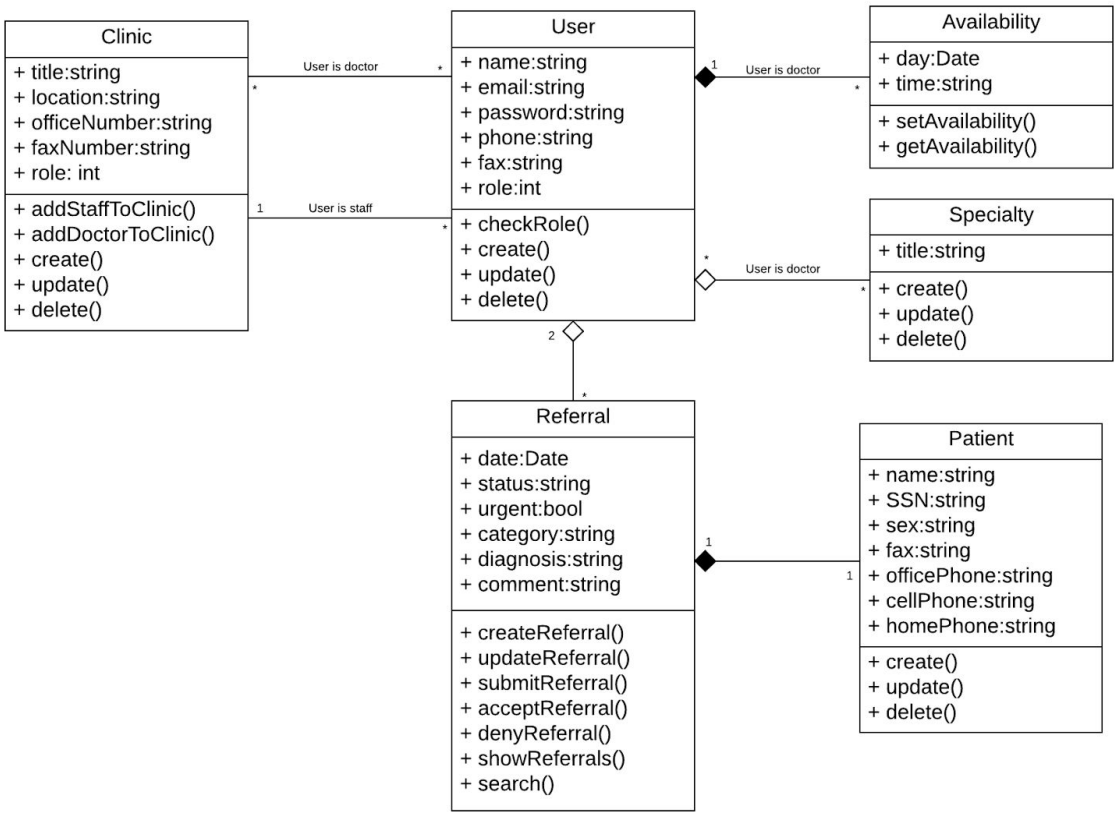
The System Architecture Diagram is a high-level overview of the fully functional NABS application, and its connections to other systems, as envisioned by the sponsor. Our prototype implementation of NABS deems some of these connections, such as IHIS (Integrated Health Information Systems) and EHR (Electronic Health Records) systems integration out of scope. These integrations, however, are consistent with the original goals of the project, and will likely be implemented in the future.

Sequence Diagram



The Sequence Diagram is a chronologically ordered representation of one full patient referral and appointment scheduling procedure using the NABS app. The involved actors (patient, users, and app) are shown, each with a “lifeline” extending from top to bottom. The top of the lifeline represents the beginning point in time, and the bottom represents the end. Thus, the vertical position of the arrows (actions) represents their chronological order in the referral and scheduling process.

Class Diagram



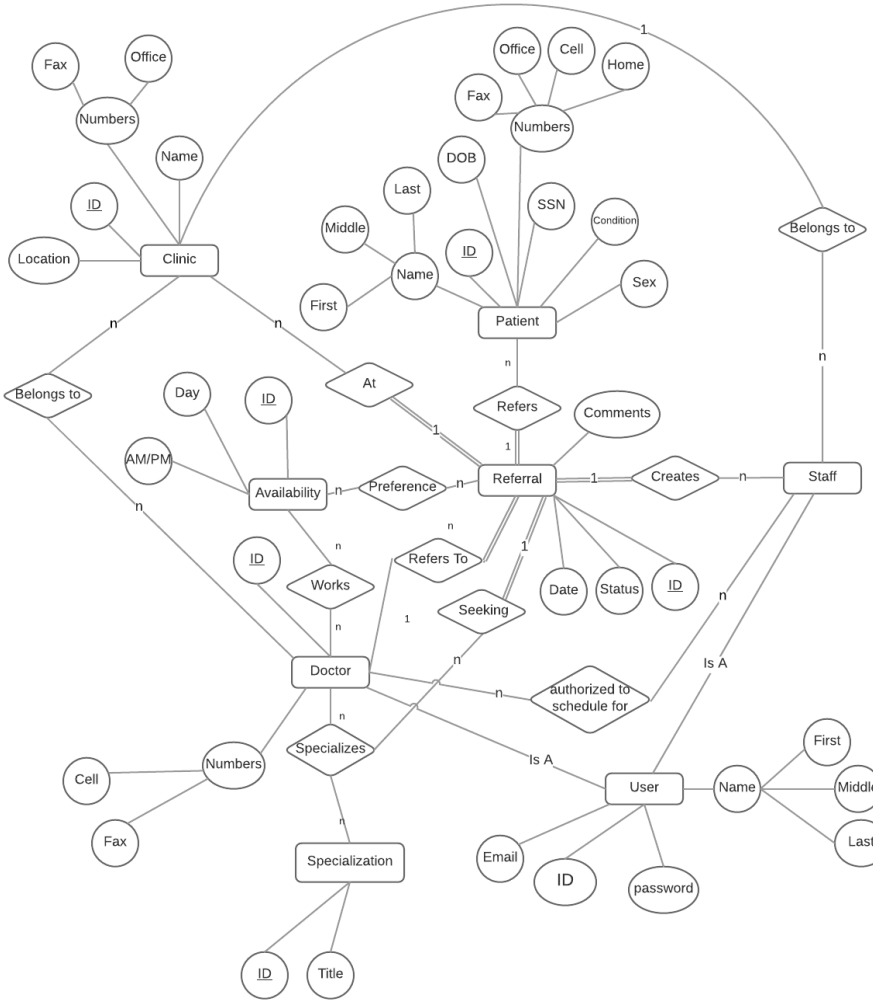
The Class Diagram describes the structure of the NABS application's implementation in terms of its classes and their relationships. It is an object-oriented model of the system. Each class has its own set of fields (top) and methods (bottom). With the reduction in scope, the availability and specialty objects were downgraded, leaving User, Clinic, and Patient as central objects of our system.

Database Design

Originally, there was a different model for each type of user, but it was changed to include a role attribute in the user model to keep track of the type of user. This allowed permissions to easily be changed only based on the role instead of having to worry about multiple models. This also allows additional roles to easily be added in the future if necessary. If a user has the doctor role, they can have specialties and availabilities. If

a user has the doctor or staff role, they can be associated with clinics. These clinics also have a role of either referred or referring so that doctors and staff can be grouped together and searched for when choosing the referring and referred doctor when making a referral. Two users with doctor roles are associated with each referral, a referring doctor and a referred doctor. Each referral contains a single patient record.

ER Diagram



The ER (Entity-Relationship) Diagram is a visual representation of the data model. The rectangular symbols are the entities of the data model, which represent the actors and objects in the system, such as Doctors. The elliptical symbols are attributes, the unique and distinguishing properties of each entity, such as the Doctor's ID and phone numbers. The diamond symbols are relationships, which describe the way in which entities interact, such as Doctors belonging to Clinics. The lines joining entities to a relationship are denoted 1 or n, which denotes whether one or many of the entity are

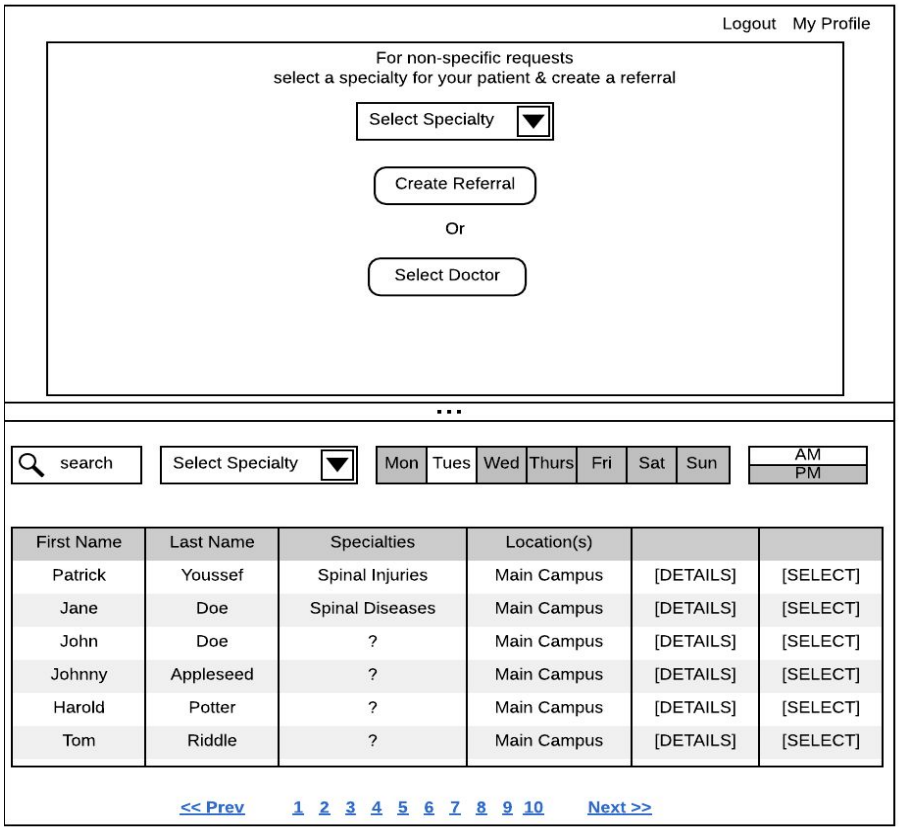
related to the other entity, and likewise the amount of the other entity involved. For instance, “Doctor belongs to Clinic” is many-to-many, because Doctors can be employed by several Clinics at one time, and each Clinic may employ several Doctors.

User Interface Design

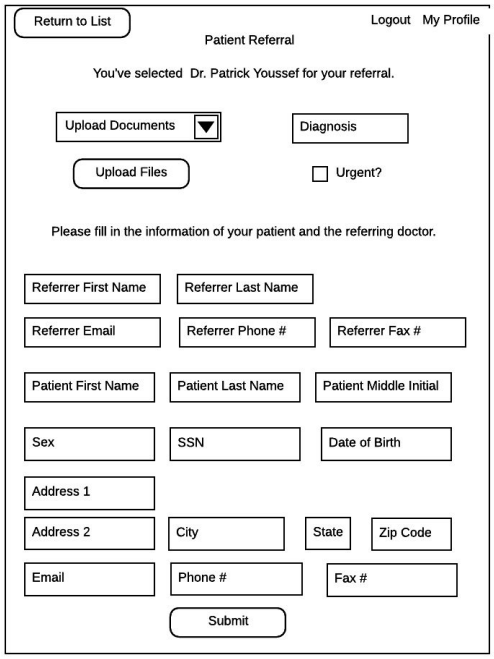
While designing the UI for NABS, we sought to create a user-friendly, intuitive, clean, and easy-to-read application, consistent with our non-functional requirements. Guided by our sponsor, we began the design process by manually drawing out preliminary mockups. Those were later used as the basis for the design of our final product, which we built using Semantic UI. The fully functional application features a wide variety of screens, often laden with entry fields, which necessitates intuitive navigation features and clear labeling.

Preliminary Mockups

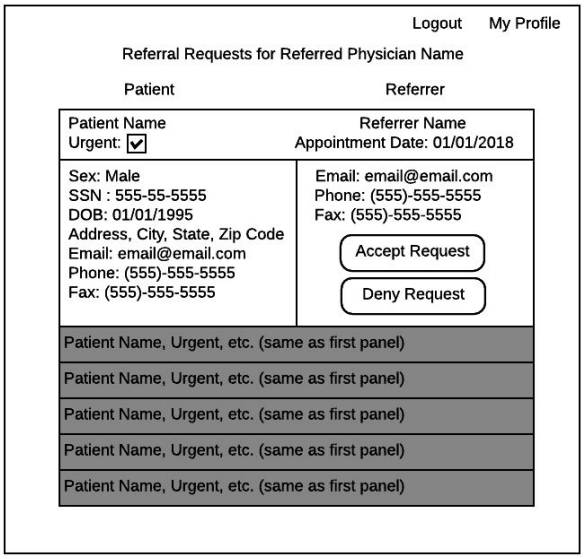
The UI design process began with the drafting of a complete set of screen mockups, in close correspondence with our sponsor. Dr. Youssef approved our mockups after several iterations, ensuring that both his and our visions were the same. The mock screens were created in early October, and were based on the specifications and scope at the time. Shown below are some of our sample mock UI screens.



Doctor Search Screen



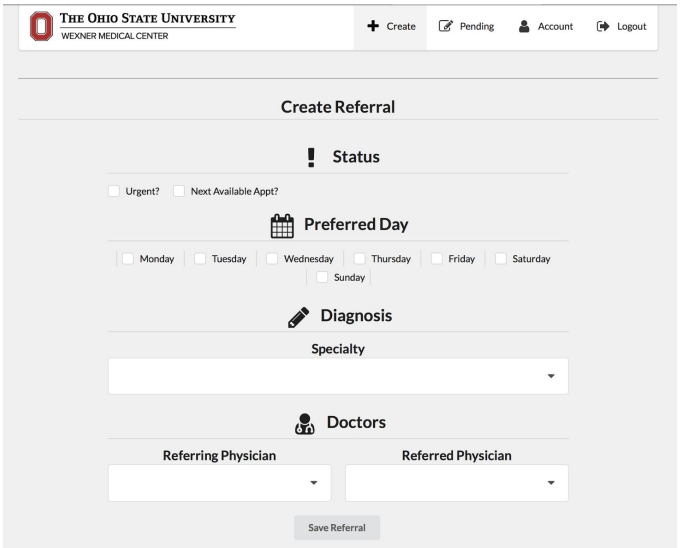
Referral Creation Screen



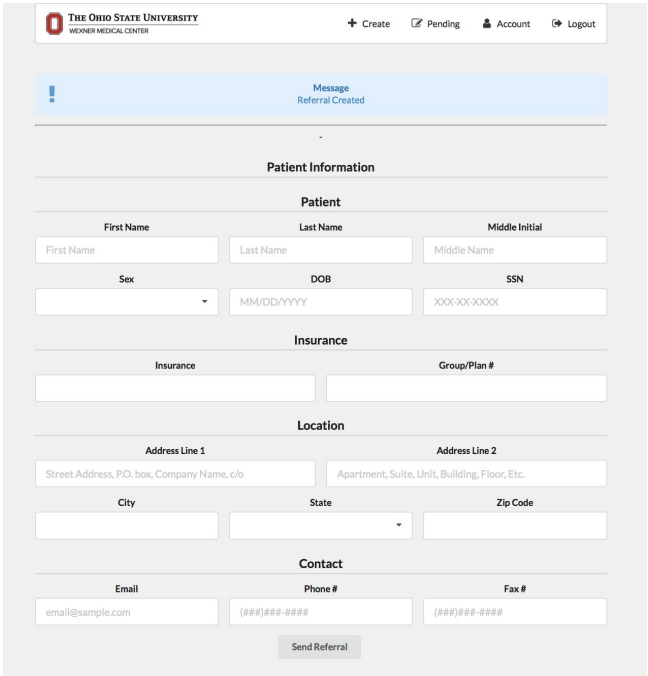
Pending Referrals Screen

Sample Screens

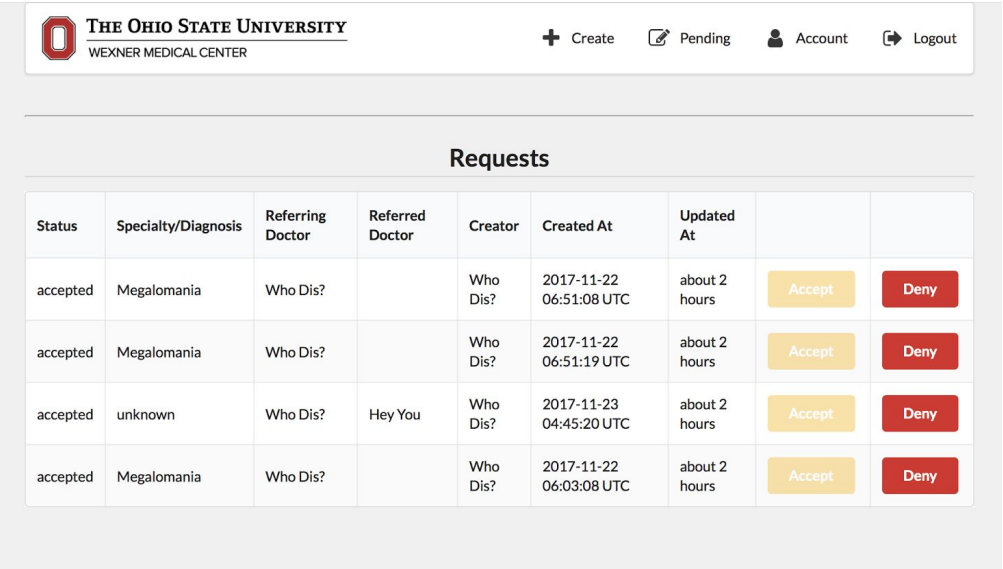
The UI of our final product has its roots in our mockups from October. However, over the two months that remained in the project's life cycle, many features were rearranged, added, or removed from scope, resulting in a largely different UI. Shown below are some sample screens from our final product.



Referral Creation Screen



Patient Information Entry Screen



Pending Referrals Screen

V. CONCLUSION

Reflection

We, as a team, successfully completed a functioning prototype for the NABS web application, implementing all important features that we had planned to since news broke of funding being halted for Ontoborn. The experience taught us the importance of assessing the project scope, which changed often in both directions throughout the project's life cycle. The most notable change in scope was precipitated by Ontoborn's withdrawal from development at the end of October, which drastically cut the number of people involved, reduced the scope, and challenged us to manage the project well enough to complete it in the five weeks that we were given. Through the use of our Trello board and by organizing regular project planning and group programming meetings, we were able to finish all that we had deemed in scope of the project.

Much of our team's success can be attributed to our strong team dynamic and distribution of strengths and weaknesses. Although team roles were loose, non-restrictive, and each member was free to contribute however they pleased, the initial roles were assigned on the basis of such strengths, and the delegation of tasks among us was simple and reliable as a result.

The Future of NABS

Our final product, ultimately, is a prototype for the web application portion of a full software suite, which will eventually include similar mobile applications for iOS and Android. We expect to pass our prototype and documentation to Ontoborn Technologies for continued development. We will remain in contact with Ontoborn beyond the semester in the event that our assistance is needed.

Within the scope of the web application alone, there are several additional features that we would like to see become a reality, such as IHIS integration, which would allow for the scheduling of patients at the clinic in one step, and transmission capabilities for patient imaging and other data through the application. It is the long-term vision of the team to see this project expanded beyond the neurosurgery department to different departments and eventually entirely different practices.

Acknowledgements

Our team would like to thank Jithendra Kumar and Agustin Ortiz from Ontoborn Technologies for their assistance throughout this project. We would also like to thank Professor Ramasamy and Professor Rajiv Ramnath for their valuable guidance, mentorship, and support.