Inpatient Therapy Scheduling System

Design Document

Team 44
Unity Point Health - Des Moines

Mathew Wymore - Team Advisor
Megan Bailey - Frontend Developer; Team Lead
Veronica Torres - Frontend Developer
Andrew Swenson - Frontend Developer
Alexis Cordts - Backend Developer
Lucas Knoll - Backend Developer
Timothy Ngo - Backend Developer

sdmay21-44@iastate.edu
https://sdmay21-44.sd.ece.iastate.edu

Version 2

Executive Summary

Development Standards & Practices Used

- IEEE Standards
- Comments within code
- Test driven development
- Branch-Review-Merge process

Summary of Requirements

- Create, read, update, delete (CRUD) therapists, nurses, patients, admin, locations
- Support multiple user types
- Ability to schedule multiple types of activities
- Ability to schedule on multiple, synced schedules
- Add/Delete rooms
- Ability to print the schedule
- Prevent scheduling conflicts
- View metrics

Applicable Courses from Iowa State University Curriculum

- SE 329 Software Project Management
- SE 339 Software Architecture and Design
- COM S 309 Software Development Practices
- COM S 319 Construction of User Interface
- COM S 363 Introduction to Database Management Systems

New Skills/Knowledge acquired that was not taught in courses

Engineering standards for software projects

Table of Contents

List of figures/tables/symbols/definitions

1 Introduction

1.1 Acknowledgement

We would like to thank Vanessa Calderon and Tara Essmann from Unity Point Clinic - Des Moines for their cooperation and support. We would also like to thank our advisor, Mathew Wymore, for his guidance and support throughout the project.

1.2 Problem and Project Statement

Problem Statement

The scheduling system for inpatient therapy at UnityPoint Health - Des Moines (hereafter referred to as The Client) should have a collaborative, intuitive interface that allows more than one user to edit schedules for a more efficient and pleasant experience for staff.

Currently The Client is using an outdated system for scheduling inpatient therapy. Their system has many error messages, and isn't intuitive for new users. Additionally, only one user can edit the schedules at a time, which leads to a lack of visibility. Their system allows patients and therapists to be double booked, and this has to be resolved manually. The current system doesn't allow the users to view the patient and therapist metrics of how much time they've spent doing each type of care. The Client needs this functionality to ensure they're following the government standards for Medicare and Medicaid. Furthermore, The Client cannot add new functionality to their current system.

Solution Approach

We will create a scheduling system in the form of a web application, which will allow for multiple users to use the system at a time, not have the error messages associated with their current system, prevent scheduling conflicts, display patient and therapist metrics, and be intuitive for new users. This will also open up opportunities for family members to view the patient's schedule as well as staff.

1.3 Operational Environment

This system is used in a healthcare environment, and must adhere to HIPPA standards. The users of the system include Therapists, Staff, and Nurses. Patients will not have access to this system. The Client will provide the hosting server for the deployed application.

1.4 Requirements

Functional Requirements

- CRUD therapists
- CRUD patients
- CRUD nurse
- CRUD admin

- CRUD location
- Add/Delete room
- Multiple users must be able to use the system at the same time
- Support multiple user types
    - Admin
    - Therapist
    - Nurse
- Ability to schedule multiple types of activities
    - Speech Therapy
    - Occupational Therapy
    - Physical Therapy
    - Conference time
    - Drive time
    - Rest time
    - Other
- Ability to schedule on multiple, synced schedules
    - View by room
    - View by therapist
- Support multiple locations
- Ability to print the schedule
- Prevent scheduling conflicts
- View metrics
    - How much time each therapist has spent doing patient care each week
    - How much time each patient has spent doing each type of therapy per day

Economic Requirements:

The budget is $0. This shouldn't be an issue since we don't have to purchase a cloud service.

Non Functional Requirements:

The application should be accessible by multiple users at once

Security Requirements:

- Must adhere to HIPPA standards
    - Patients can't view other patient information
- Nurses can't alter the schedule, they can only view it
- Sensitive information (passwords) must be hashed before storage

1.5 Intended Users and Uses

Administrators (staff) will have the ability to Create, Read, Update, Delete (CRUD) other admins, CRUD locations for therapy, CRUD nurses in the system, CRUD appointments for patients and therapists, CRUD patients in the system. Additionally, admins will be able to view the metrics of how much time patients and therapists have spent doing each type of therapy per week, and view the schedule of appointments. They will also be able to add and delete therapy types, therapy subtypes, and rooms for each location.

Therapists will have the ability to CRUD appointments for patients and their own schedule, and CRUD patients in the system. Additionally, therapists will be able to view the metrics of how much time patients and therapists have spent doing each type of therapy per week, and view the schedule of appointments.

Nurses will only be able to view the schedule of appointments. Patients are not users of our system.

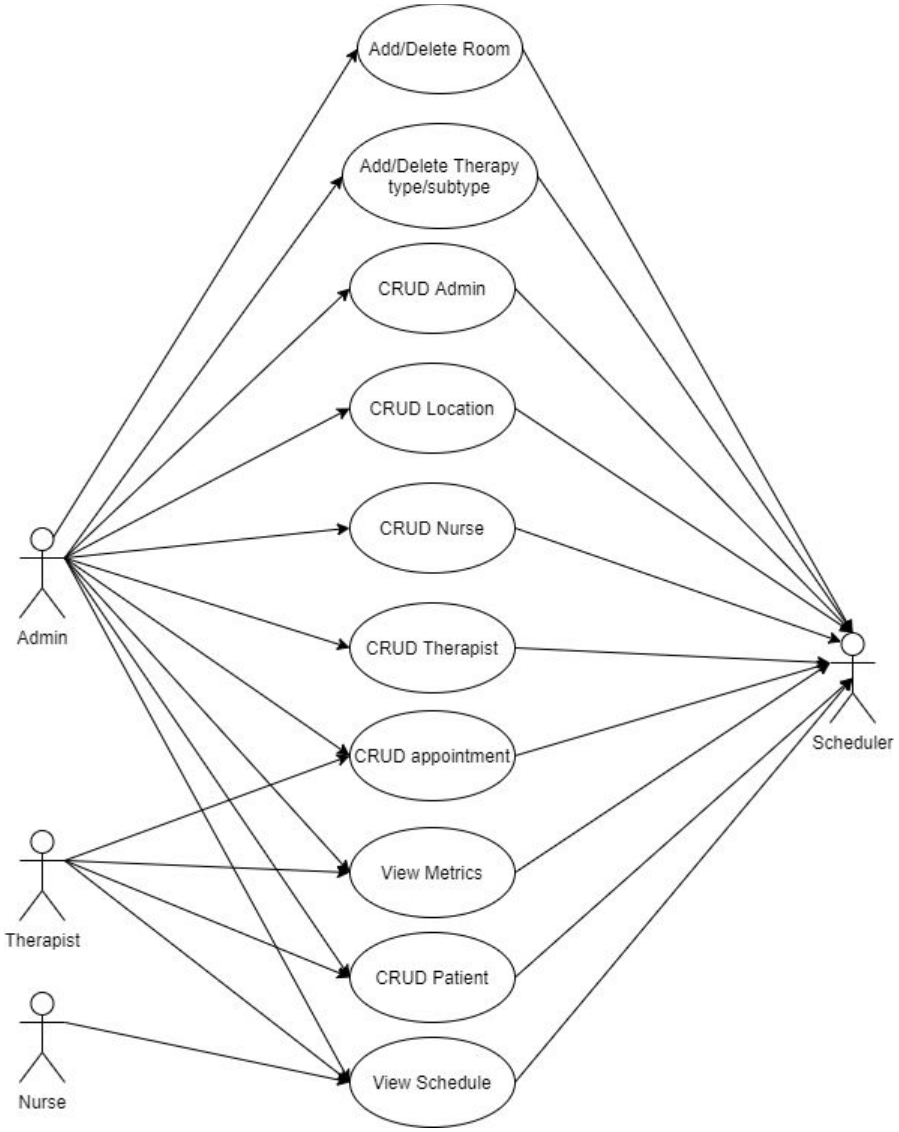


Figure 1.1 Use Case Diagram

1.6 Assumptions and Limitations

Assumptions

- Blue is The Client's logo color and will be used as the primary UI color
- Max number therapists: 50
- Max number of patients: 50
- Max number of nurses: 10
- Max number of admins: 6
- Maximum number of location: 2
- Max number of rooms: 49
- The Client will only need to schedule out 1 week in advance
- Users will be connected to the internet while using the application

Limitations

- The cost of the product should not exceed $0
- The system should work with Google Chrome version 85.0 and Motzilla Firefox version 80.0 and Microsoft Edge version 85.0

1.7 Expected End Product and Deliverables

Inpatient Therapy Scheduling System, Delivered by May 1, 2021

The inpatient Therapy Scheduling System will be a web application adhering to the requirements listed above (see section 1.4). The code will be delivered in the form of a GitHub repository, with documentation throughout the code base for readability and maintainability.

2 Project Plan

2.1 Task Decomposition

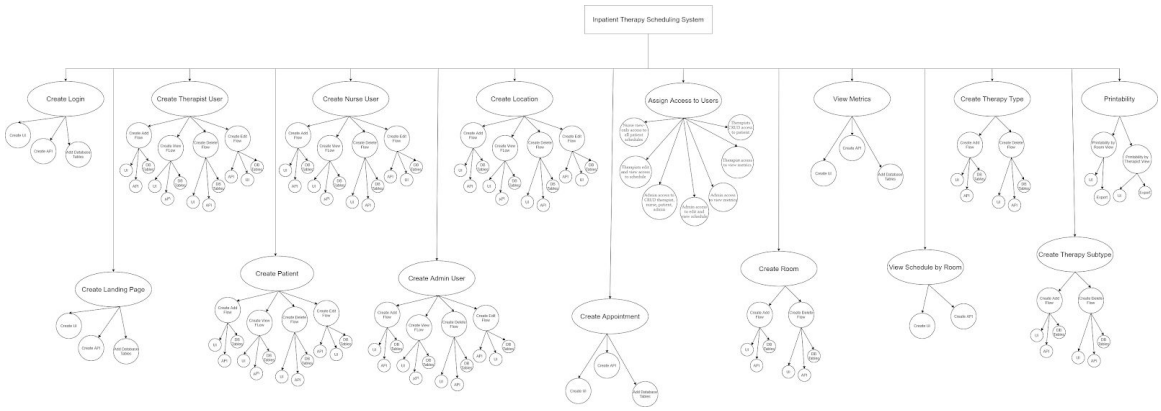


Figure 2.1 Task Decomposition Diagram

1. Create the login flow
    a. UI
        i. Appropriate error messages
    b. DB tables
    c. API endpoints
2. Create the landing page
    a. UI of schedule view
    b. API endpoints for schedule
    c. DB tables for schedule
3. Create Therapist User
    a. Create the Add Therapist flow
        i. UI
        ii. API endpoints
        iii. DB tables
    b. Create the View Therapist flow
        i. UI
        ii. API endpoints
        iii. DB tables
    c. Create the Delete Therapist flow
        i. UI
        ii. API endpoints
        iii. DB tables
    d. Create the Update Therapist flow
        i. UI
        ii. API endpoints
        iii. DB tables
4. Repeat 3 - 6 for Admin user
5. Repeat 3 - 6 for Patient

6. Repeat 3 - 6 for Nurse user
7. Create Location
    a. Create the Add Location flow
        i. UI
        ii. API endpoints
        iii. DB tables
    b. Create the Delete Location flow
        i. UI
        ii. API endpoints
        iii. DB tables
    c. Create the View Location flow
        i. UI
        ii. API endpoints
        iii. DB tables
    d. Create the Edit Location flow
        i. UI
        ii. API endpoints
        iii. DB tables
8. Create Room
    a. Create the Add Room flow
        i. UI
        ii. API endpoints
        iii. DB tables
    b. Create the Delete Room flow
        i. UI
        ii. API endpoints
        iii. DB tables
9. Create Therapy Type
    a. Create the Add Therapy Type flow
        i. UI
        ii. API endpoints
        iii. DB tables
    b. Create the Delete Therapy Type flow
        i. UI
        ii. API endpoints
        iii. DB tables
10. Create Therapy Subtype
    a. Create the Add Therapy Subtype flow
        i. UI
        ii. API endpoints
        iii. DB tables
    b. Create the Delete Therapy Subtype flow
        i. UI
        ii. API endpoints
        iii. DB tables
11. Create Appointment
    a. Create the Create Appointment flow

        i. UI
        ii. API endpoints
        iii. DB tables
    b. Create the View Appointment flow
        i. UI
        ii. API endpoints
        iii. DB tables
    c. Create the Delete Appointment flow
        i. UI
        ii. API endpoints
        iii. DB tables
    d. Create the Edit Appointment flow
        i. UI
        ii. API endpoints
        iii. DB tables
    e. Save the appointment details for metric viewing later
12. Create the View Metric flow
    a. UI
    b. API endpoints
    c. DB tables
13. View schedule by room
    a. UI
    b. API endpoints
14. Printability
    a. Allow printability for room schedule view
        i. UI
        ii. Export to PDF or Excel
    b. Allow printability for therapist schedule view
        i. UI
        ii. Export to PDF or Excel
15. Assign access to users
    a. Give nurse view-only access to all patient schedules
    b. Give therapists edit and view access to schedule
    c. Give therapists CRUD access to patient
    d. Give therapists access to view metrics
    e. Give admin access to CRUD therapist, nurse, patient, admin
    f. Give admin access to view metrics
    g. Give admin access to edit and view schedule

2.2 Risks And Risk Management/Mitigation

We evaluated the risks associated with this software project in Table 2.2. We determined the risk probability using Table 2.1.

High Likelihood	3	4	5
Med Likelihood	2	3	4

Low Likelihood	1	2	3
	Low Impact	Med Impact	High impact

Table 2.1 Risk Probability Definition

RISK	RISK PROBABILITY	MITIGATION STRATEGY
HIPPA compliance	3	- Don't use real data when developing - Only deploy on their local server so you can't access it outside of their network
Login security	3	- Hash sensitive information - Enforce password requirements
Project timeline flaws	2	
Requirements inflation	4	- Communicate clearly early on and throughout the development process - Verify the scope with the stakeholders - Create a clear project schedule
Maintainability	1	
Poor integration among tech stack	2	
Low stakeholder engagement	1	
Poor user experience	1	
Teammate turnover	1	
Requirements misinterpreted	3	- Demo often to client - Follow design documents
Low productivity	1	

Table 2.2 Risk Analysis

2.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

1. Login
    a. A user can login and is redirected to landing page

    b. Password will be hashed and stored
2. Create users
    a. Therapist user can be created
    b. Nurse user can be created
    c. Patient can be created
    d. Admin user can be created
3. Usable interface
    a. Therapist can add appointment
    b. Appointment can be viewed in the UI
4. Calculable metrics
    a. Metrics are tallied and calculated
    b. Metrics are displayed
5. Have MVP complete

2.4 Project Timeline/Schedule

In Table 2.3 you'll find a detailed schedule of the project timeline. The delivery dates correspond to our biweekly meetings with The Client, where we will demo the project's progress and receive feedback.

Our progress will be made from October 1, 2020 through November 7, 2020. This will leave us time before the semester ends to create final presentations for SE 491 and The Client. Then we will cease work for winter break. We will finish the project between January 25, 2021 and April 8, 2021. This will give us a few weeks for slack time and to create final presentations for SE 492 and The Client.

For a combined chart/schedule view of the Gantt chart, see Appendix A.

TASK NAME	START DATE	END DATE	DELIVERY DATE	START ON DAY	DURATION (WORK DAYS)
Create Login Flow			10/21/2020		
Create the DB tables for login	10/1	10/2		0	1
Create the UI for login	10/1	10/5		0	4
Create the API endpoints for login	10/2	10/6		1	4
Handle incorrect login	10/7	10/8		6	1
Handle successful login	10/7	10/8		6	1
Create the Landing Page			10/21/2020		
Create UI of schedule view	10/8	10/15		7	7
Create API endpoints for schedule	10/6	10/14		5	8
Create the DB tables for schedules	10/6	10/7		5	1
Create Therapist User			11/4/2020		

Create the Add Therapist flow	10/15	10/25		14	17
Create the View Therapist flow	10/25	10/29		31	4
Create the Delete Therapist flow	10/29	11/1		35	4
Create the Update Therapist flow	11/1	11/4		40	3
Create Admin User			2/17/2021		
Create the Add Admin flow	1/25	1/30		116	5
Create the View Admin flow	1/30	2/2		121	3
Create the Delete Admin flow	2/2	2/5		124	3
Create the Update Admin flow	2/5	2/8		127	3
Create Nurse User			3/3/2021		
Create the Add Nurse flow	2/15	2/18		137	3
Create the View Nurse flow	2/20	2/23		142	3
Create the Delete Nurse flow	2/23	2/26		145	3
Create the Update Nurse flow	2/26	3/1		148	3
Create Patient			3/17/2021		
Create the Add Patient flow	2/8	2/13		130	5
Create the View Patient flow	2/13	2/18		135	5
Create the Delete Patient flow	2/18	2/23		140	5
Create the Update Patient flow	2/23	2/28		145	5
Create Location			3/17/2021		
Create the Add Location flow	3/1	3/8		151	7
Create the View Location flow	3/8	3/11		158	3
Create the Delete Location flow	3/11	3/13		161	2
Create the Update Location flow	3/13	3/16		163	3
Create Room			3/17/2021		
Create the Add Room flow	2/28	3/3		150	3
Create the Delete Room flow	3/3	3/4		153	1
Create Therapy Type			3/31/2021		
Create the Add Therapy Type flow	3/16	3/20		166	4
Create the Delete Therapy Type	3/20	3/22		170	2

Flow					
Create Therapy Subtype			3/31/2021		
Create the Add Therapy Subtype flow	3/19	3/21		169	2
Create the Delete Therapy Subtype Flow	3/21	3/23		171	2
Create Appointment			4/14/2021		
Create the Add Appointment flow	3/23	3/31		173	8
Create the View Appointment flow	3/31	4/3		181	3
Create the Delete Appointment flow	4/3	4/6		184	3
Create the Edit Appointment flow	4/6	4/9		187	3
Create the View Metric flow			4/14/2021		
Create the View Metric UI	3/31	4/5		181	5
Create View Metric DB Tables	3/31	4/1		181	1
Create the API endpoints	4/1	4/4		182	3
Add Printability			4/14/2021		
Create the Printing UI	4/5	4/9		186	4
Create the Printing API endpoints	4/5	4/8		186	3
Assign Access to Users			3/3/2021		
Give nurse view-only access to schedules	3/1	3/2		151	1
Give therapists edit and view access to schedule	11/6	11/7		45	1
Give therapists CRUD access to patient	11/6	11/7		45	1
Give therapists access to view metrics	11/6	11/7		45	1
Give admin access to CRUD therapist, nurse, patient, admin	2/8	2/10		130	2
Give admin access to edit and view schedule	2/8	2/9		130	1

Table 2.3 Project Timeline

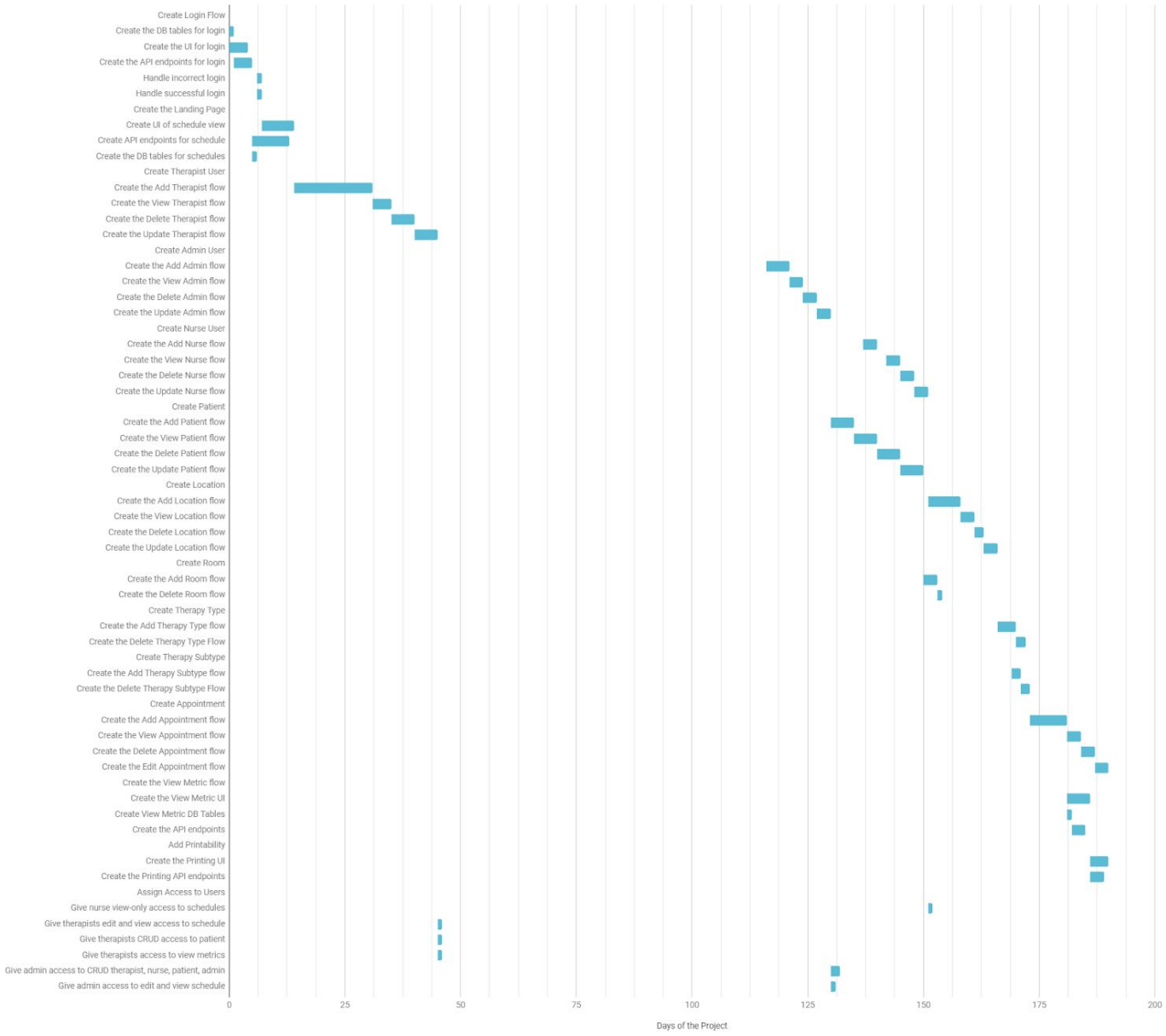


Figure 2.2 Gantt Chart

2.5 Project Tracking Procedures

We'll be using GitHub issues to define and track tasks of the project. This will allow high visibility for everyone to see relevant information about the task. We'll use the branch-review-merge procedure to ensure that code adheres to the level of quality we expect from our team.

2.6 Personnel Effort Requirements

We've calculated the personnel effort requirements by personnel hours in Table 2.4. If two developers are working together on a task, then each hour they work together equates to two personnel hours.

TASK NAME	PERSONNEL HOURS
Create Login Flow	
Create the DB tables for login	3
Create the UI for login	9
Create the API endpoints for login	10
Handle incorrect login	2
Handle successful login	2
Create the Landing Page	
Create UI of schedule view	14
Create API endpoints for schedule	14
Create the DB tables for schedules	4
Create Therapist User	
Create the Add Therapist flow	22
Create the View Therapist flow	8
Create the Delete Therapist flow	8
Create the Update Therapist flow	8
Create Admin User	
Create the Add Admin flow	8
Create the View Admin flow	6
Create the Delete Admin flow	6
Create the Update Admin flow	6
Create Nurse User	

Create the Add Nurse flow	6
Create the View Nurse flow	6
Create the Delete Nurse flow	6
Create the Update Nurse flow	6
Create Patient	
Create the Add Patient flow	10
Create the View Patient flow	10
Create the Delete Patient flow	10
Create the Update Patient flow	10
Create Location	
Create the Add Location flow	14
Create the View Location flow	6
Create the Delete Location flow	4
Create the Update Location flow	6
Create Room	
Create the Add Room flow	6
Create the Delete Room flow	1
Create Therapy Type	
Create the Add Therapy Type flow	6
Create the Delete Therapy Type Flow	3
Create Therapy Subtype	
Create the Add Therapy Subtype flow	4
Create the Delete Therapy Subtype Flow	2
Create Appointment	
Create the Add Appointment flow	16
Create the View Appointment flow	6
Create the Delete Appointment flow	6
Create the Edit Appointment flow	6
Create the View Metric flow	
Create the View Metric UI	10

Create View Metric DB Tables	2
Create the API endpoints	6
Add Printability	
Create the Printing UI	8
Create the Printing API endpoints	5
Assign Access to Users	
Give nurse view-only access to schedules	1
Give therapists edit and view access to schedule	1
Give therapists CRUD access to patient	2
Give therapists access to view metrics	1
Give admin access to CRUD therapist, nurse, patient, admin	2
Give admin access to edit and view schedule	1
TOTAL	**319**

Table 2.4 Personnel Requirements

2.7 Other Resource Requirements

- GitHub
- GitHub issues
- ISU server from ETG

2.8 Financial Requirements

No financial resources will be required for this project. We are students, so we have access to Github for free, and we will be doing development on an ISU server at no cost.

3 Design

3.1 Previous Work And Literature

We spent some time looking into existing scheduling applications within the healthcare industry and found Shift Admin, QGenda, and Care Cloud to be highly rated [1]. Shift Admin has a mobile app. Our application will be viewable on mobile devices through a mobile browser. QGenda does automated scheduling. The Client doesn't want this, as they want their physicians to control their own schedules. Care Cloud has a chat bot. This feature wouldn't generate much value for The Client since their patients aren't scheduling their own appointments.

The main difference between these products and ours is that ours will be free for The Client to use.

3.2 Design Thinking

During the "define" stage we met with The Client and discussed problems with the old application to define the shortcomings, then we discussed requirements as a team and got the requirements approved by The Client. Some of their problems included only one user being able to interact with the application at a time; there were broken parts of the application that no one knew how to fix; the user experience wasn't intuitive for new users, which led to a lot of training; Because they didn't have great visibility into the application all the time, they had to print out the schedules multiple times per day; the metric calculation features were broken; they weren't storing their data anywhere.

We considered creating a desktop application but the client wants the application to be visible by multiple users at the same time. We decided a web application would allow for greater visibility for therapists to use on tablets and mobile devices.

3.3 Proposed Design

Our plan is to create a web application that holds the schedule and key information and functionality that goes with it. We will have a user structure that goes as follows; Admin, Therapist, Nurse. Each of these roles has access to different parts of the website. From viewing different schedule views, creating new users, scheduling appointments, viewing therapist and patient metrics, as well as other features. We plan to have multiple ways to view the schedule to increase the customization options users have.

The UI design adheres to their color and typography scheme of The Client. The current design excels in giving the users high visibility, ease of access, as well as a more intuitive user interface. It matches the core functionality their previous system uses but adds on more features that can improve user satisfaction, such as:

- The ability to view metrics of how much time therapists have spent performing each type of therapy with patients
- The ability to prevent scheduling conflicts
- Easily add and remove rooms and locations
- Allow nurses to have access to the system

3.4 Technology Considerations

The technologies we chose are currently maintained and have ample documentation available. The tech stack we chose is very common, so a technologist working for The Client should be able to maintain the application.

We chose React instead of Angular because the scheduling system is more component based, which is a strength of React. We chose SQL over NoSQL database because we wanted a relational database since a lot of the data for our system is interconnected and it gives us more flexibility in how we create the system. We chose Javascript over Typescript because we had more experience with Javascript, even though Typescript has type checking built in.

3.5 Design Analysis

Our design discussed in 3.3 has worked so far. Our web application route means we have many examples of applications to base our design off of. We haven't ran into any problems with the technology we're using (React, Javascript, SQL, Java).

We will continue to get feedback from The Client through demos and modify our design and structure of the web application as needed.

3.6 Development Process

We are following Test Driven Development (Figure 3.1) and Agile methodologies (Figure 3.2). We are using Test Driven Development to minimize bugs in the application, ensure high code coverage, and deliver high quality code. We are using an Agile methodology that is structured similarly to Kanban. We are demoing our project to The Client every two weeks (at the end of our sprints) and receiving feedback from The Client at that time. We wanted to receive feedback from The Client throughout the development process so we could adjust the project as needed throughout the development process instead of changing things last minute.

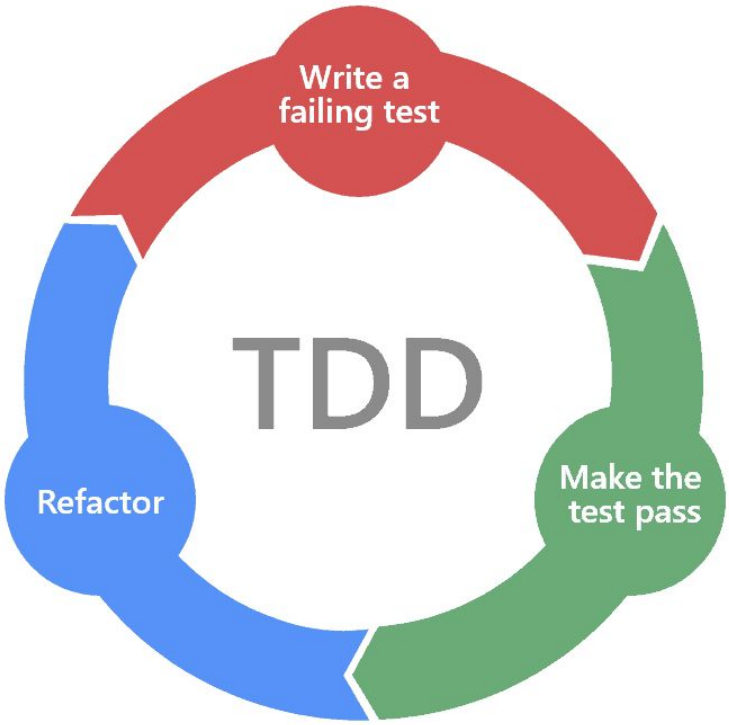


Figure 3.1 Test Driven Development

Source: [2]

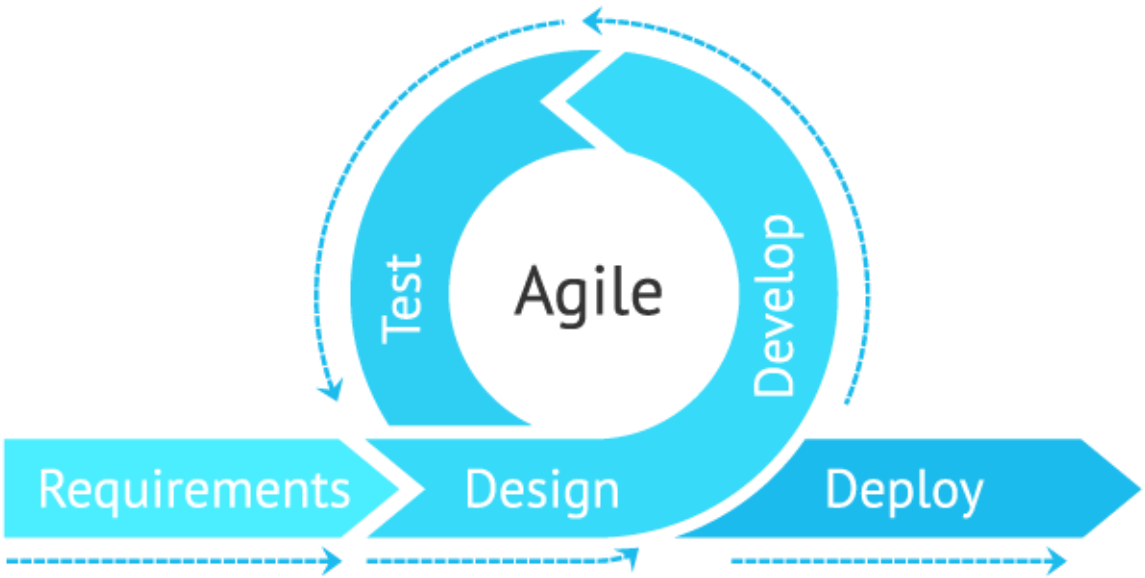


Figure 3.2 Agile Process

Source: [3]

3.7 Design Plan

Our design plan was formulated around our use-cases we laid out in section 1.5 and our requirements. We wanted to construct our application around the needs of the user. We modeled features, settings and even pages completely around the user to ensure a friendly system. We heavily integrated each use case into a main part of our software to ensure each use case's requirements were met. Below we have our block diagram which shows the general flow of our UI from an Admin users perspective. We also have our Software Architecture Overview displaying how the system will be built out from the front end UI to the backend system. If you look carefully at each diagram our use cases and other requirements are the focal point of our software.

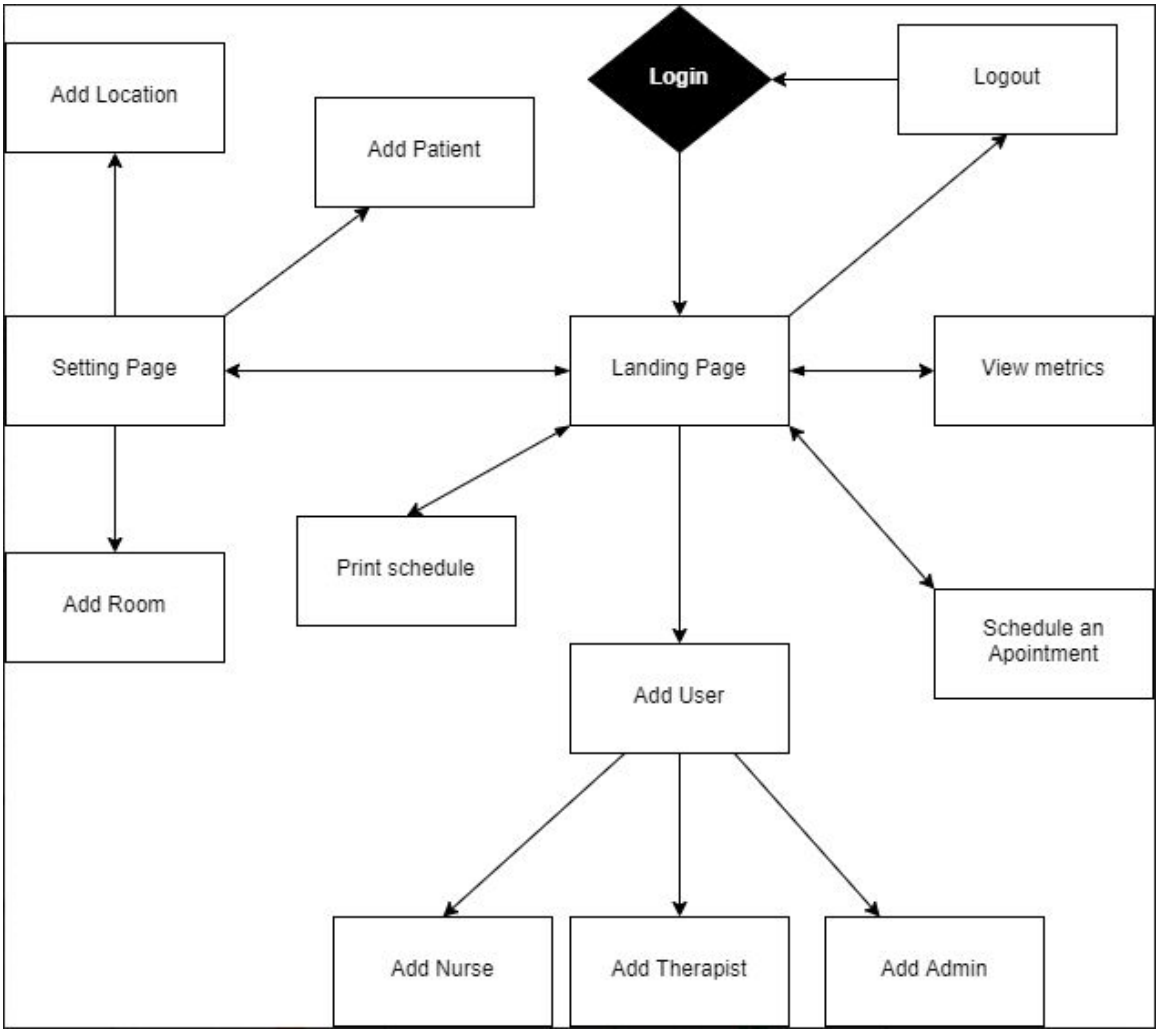


Figure 3.3 Block Diagram

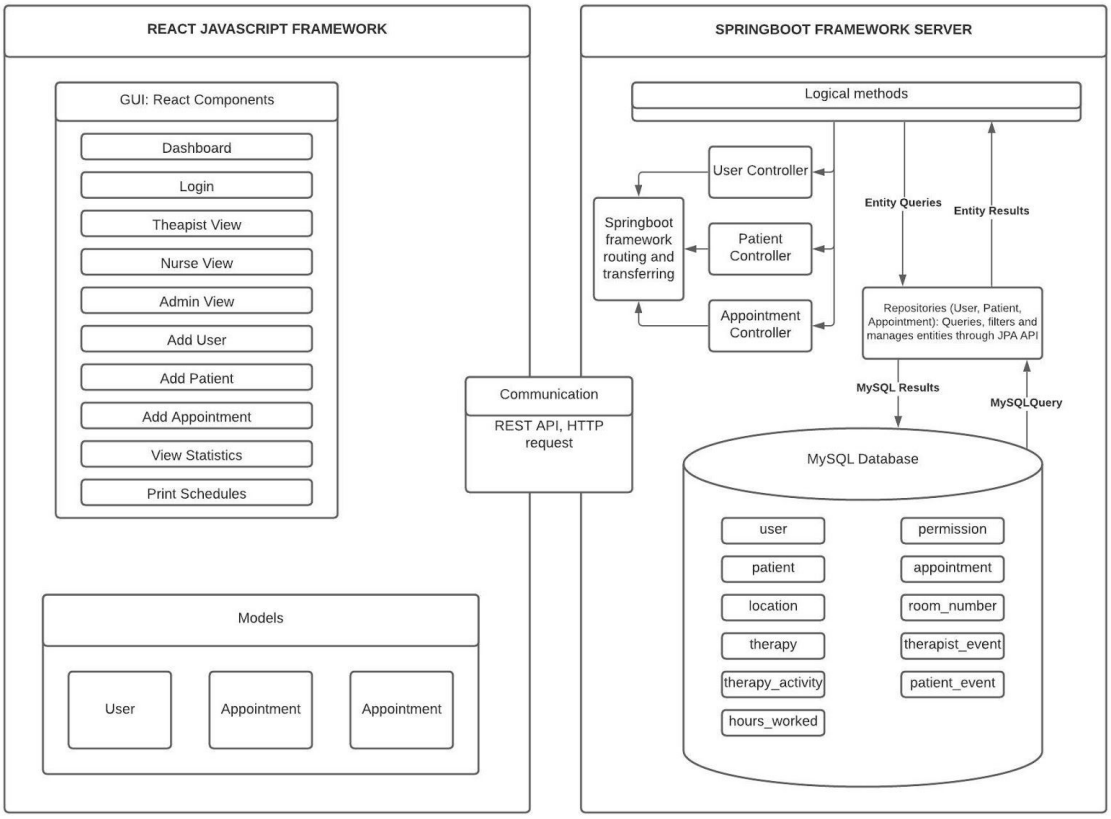


Figure 3.4 Software Architecture Overview

4 Testing

4.1 Unit Testing

There are no hardware components to this project, so unit testing will only be performed on software. Following Test Driven Development, developers will write unit tests as they code. Unit tests will be written using React Testing Library for the front end and JUnit and Mockito for the backend. Each area of the project will be unit tested:

- Login
- CRUD therapist
- CRUD patient
- CRUD admin
- CRUD nurse
- CRUDappointment
- Create/Delete therapy type
- Create/Delete therapy subtype
- Create/Delete location
- Create/Delete room

- Print schedules

4.2 Interface Testing

Developers will manually test the interface of the application. The areas of the interface to be manually tested are:

- Login
- view schedule by therapist
- view schedule by room
- create/delete an appointment
- create/delete location
- create/delete room
- create/delete therapy type and subtype
- CRUD therapist
- CRUD patient
- CRUD admin
- CRUD nurse
- CRUDappointment
- Create/Delete therapy type
- Create/Delete therapy subtype
- Create/Delete location
- Create/Delete room
- Print schedules

4.3 Acceptance Testing

For acceptance testing, we will ensure 100% of tests are passing successfully and demo the application to The Client every two weeks. The Client will have final say on if the application is accepted or not.

4.4 Results

When demonstrating our initial UI to The Client we received feedback on our initial designs. The Client was able to further explain some of the requirements. We learned that Nurses don't need to see both schedule views - they only need to see the room schedule view. We also found that Nurses are the users that need to see the therapy subtype for each patient, so that can be a smaller part of the schedule than we originally thought.

We plan to address these comments and incorporate the changes into our application before moving on to the next task. This ensures The Client's comments and feedback don't get overlooked.

5 Implementation

Describe any (preliminary) implementation plan for the next semester for your proposed design in 3.3.

6 Closing Material

6.1 Conclusion

Summarize the work you have done so far. Briefly reiterate your goals. Then, reiterate the best plan of action (or solution) to achieving your goals and indicate why this surpasses all other possible solutions tested.

6.2 References

[1] "Best Medical Staff Scheduling Software," G2. [Online]. Available: https://www.g2.com/categories/medical-staff-scheduling. [Accessed: 22-Sep-2020].

[2] Test Driven Development [Online image]. Why Test-Driven Development (TDD). https://marsner.com/blog/why-test-driven-development-tdd/. [Accessed: 20-Oct-2020].

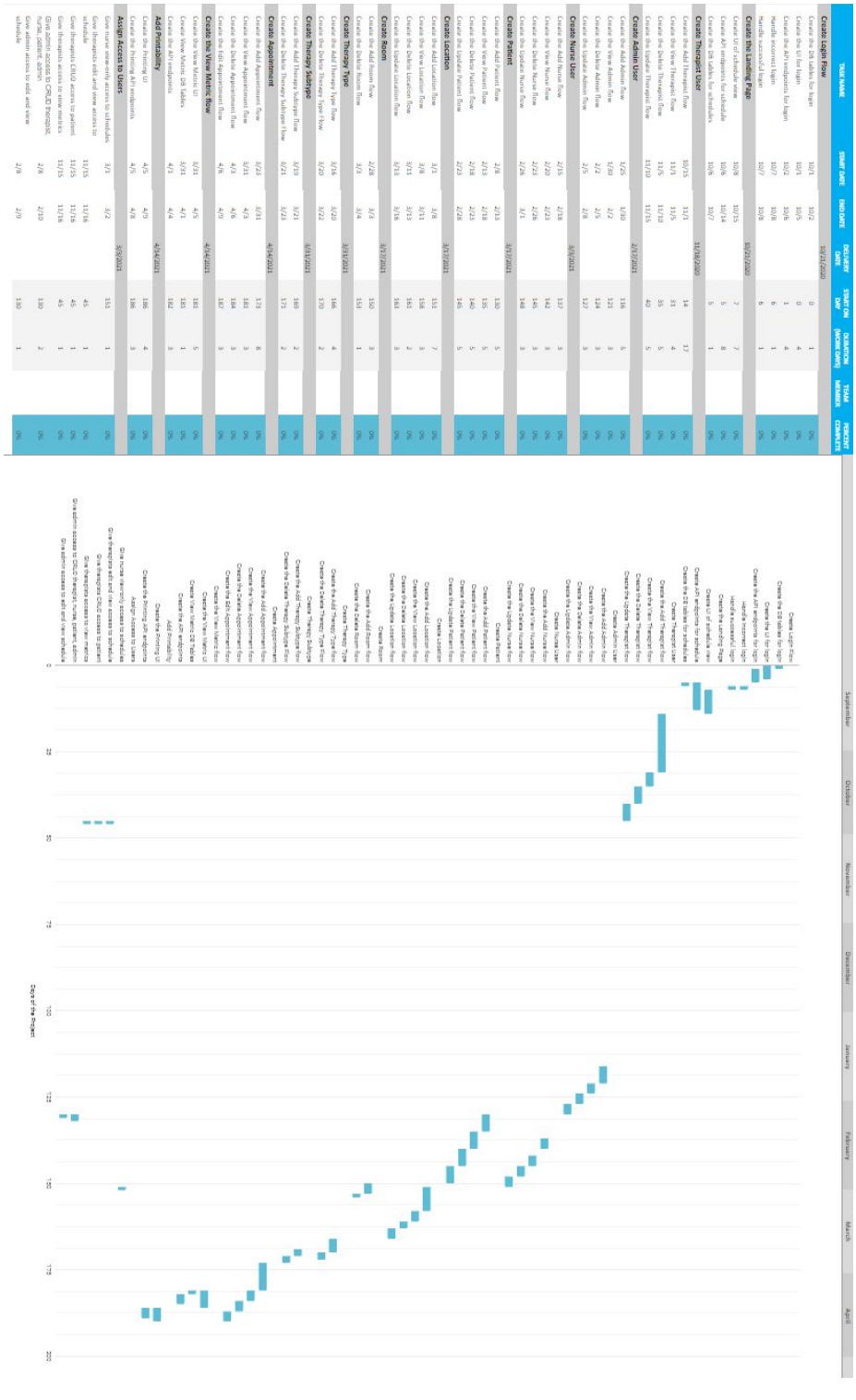
[3] Agile Process [Online image]. DevCom. https://devcom.com/tech-blog/agile-advantages-for-business/. [Accessed: 20-Oct-2020].

6.3 Appendices

Appendix A: Gantt Chart

Appendix A

Gantt Chart



TASK NAME	START DATE	END DATE	DEADLINE DATE	START ON DAY	DURATION (WORK DAYS)	TEAM MEMBER	PERCENT COMPLETE
Create Login Flow	10/1	10/2	10/2/2020	0	1		0%
Create the DB tables for login	10/1	10/2		0	1		0%
Create the UI for login	10/1	10/5		0	4		0%
Create the API endpoints for login	10/2	10/6		1	4		0%
Handle incorrect login	10/7	10/8		6	1		0%
Handle successful login	10/7	10/8		6	1		0%
Create the Landing Page	10/8	10/15	10/21/2020	7	7		0%
Create UI of schedule view	10/8	10/14		7	7		0%
Create API endpoints for schedules	10/8	10/14		7	7		0%
Create the DB tables for schedules	10/8	10/7		5	1		0%
Create Therapist User	10/15	11/1	11/16/2020	14	17		0%
Create the Add Therapist Flow	10/15	11/1		14	17		0%
Create the View Therapists Flow	11/1	11/5		31	4		0%
Create the Delete Therapist Flow	11/5	11/10		35	5		0%
Create the Update Therapist Flow	11/10	11/15		40	5		0%
Create Admin User	2/17/2021						0%
Create the Add Admin Flow	1/25	1/30		116	5		0%
Create the View Admin Flow	1/30	2/2		121	3		0%
Create the Update Admin Flow	2/2	2/5		124	3		0%
Create the Delete Admin Flow	2/5	2/8		127	3		0%
Create Nurse User	2/15	2/18	3/3/2021	137	3		0%
Create the Add Nurse Flow	2/15	2/18		137	3		0%
Create the View Nurse Flow	2/20	2/23		142	3		0%
Create the Delete Nurse Flow	2/23	2/26		145	3		0%
Create the Update Nurse Flow	2/26	3/1		148	3		0%
Create Patient	2/8	2/13	3/17/2021	130	5		0%
Create the Add Patient Flow	2/8	2/13		130	5		0%
Create the View Patient Flow	2/13	2/18		135	5		0%
Create the Delete Patient Flow	2/18	2/23		140	5		0%
Create the Update Patient Flow	2/23	2/28		145	5		0%
Create Location	3/1	3/8	3/17/2021	153	7		0%
Create the Add Location Flow	3/1	3/8		153	7		0%
Create the View Location Flow	3/8	3/11		158	3		0%
Create the Delete Location Flow	3/11	3/13		161	2		0%
Create the Update Location Flow	3/13	3/16		163	3		0%
Create Room	2/26	3/3	3/17/2021	150	3		0%
Create the Add Room Flow	2/26	3/3		150	3		0%
Create the Delete Room Flow	3/1	3/4		153	1		0%
Create Therapy Type	3/16	3/20	3/31/2021	166	4		0%
Create the Add Therapy Type Flow	3/16	3/20		166	4		0%
Create the Delete Therapy Type Flow	3/20	3/22		170	2		0%
Create Therapy Subtype	3/20	3/21	3/31/2021	170	2		0%
Create the Add Therapy Subtype Flow	3/20	3/21		170	2		0%
Create the Delete Therapy Subtype Flow	3/21	3/23		171	2		0%
Create Appointment	3/23	3/31	4/14/2021	173	8		0%
Create the Add Appointment Flow	3/23	3/31		173	8		0%
Create the View Appointment Flow	3/31	4/3		181	3		0%
Create the Delete Appointment Flow	4/1	4/6		184	3		0%
Create the Edit Appointment Flow	4/6	4/9		187	3		0%
Create the View Metric flow	3/31	4/5	4/14/2021	181	5		0%
Create the View Metric UI	3/31	4/1		181	1		0%
Create View Metric DB Tables	3/31	4/4		181	4		0%
Create the API endpoints	4/1	4/4		182	3		0%
Add Printability	4/5	4/9	4/14/2021	186	4		0%
Create the Printing UI	4/5	4/9		186	4		0%
Create the Printing API endpoints	4/5	4/8		186	3		0%
Assign Access to Users	3/31	4/2	4/14/2021	153	1		0%
Give nurse read-only access to schedules	3/31	4/2		153	1		0%
Give therapists edit and view access to schedule	11/15	11/16		45	1		0%
Give therapists CRUD access to patient	11/15	11/16		45	1		0%
Give therapists access to view metrics	11/15	11/16		45	1		0%
Give admin access to CRUD therapist, nurse, patient, admin	2/8	2/10		130	2		0%
Give admin access to edit and view schedule	2/8	2/9		130	1		0%