

UNIVERSITY OF CALGARY  
DEPARTMENT OF COMPUTER SCIENCE  
CPSC 481, FALL 2022

---

## Calcare Portfolio

---



### Group #6

Ricky Bhatti (ricky.bhatti1@ucalgary.ca)  
Quyanna Campbell (quyanna.campbell@ucalgary.ca)  
Farhan Ahmed (syed.ahmed6@ucalgary.ca)  
Haris Ahmad (haris.ahmad1@ucalgary.ca)  
Vardhini Pawar (vardhini.pawar@ucalgary.ca)

October 17, 2022

## Contents

<b>1 Phase #0: Setting the stage</b>	<b>1</b>
1.1 Background Environment . . . . .	1
1.2 What the system will be used for as well as the general expectations that the system should satisfy . . . . .	1
1.3 System Constraints . . . . .	1
<b>2 Phase #1: Identification</b>	<b>2</b>
2.1 Expected Types of Users . . . . .	2
2.2 Work Contexts . . . . .	2
2.3 Concrete Task Examples . . . . .	2
<b>3 Phase #2: Tentative list of requirements</b>	<b>4</b>
3.1 Calcare Requirements . . . . .	4
3.1.1 Must Include: . . . . .	4
3.1.2 Should Include: . . . . .	4
3.1.3 May Include: . . . . .	4
3.1.4 Exclude: . . . . .	5
<b>4 Phase #3: Prototyping</b>	<b>6</b>
4.1 Prototype #1 (Appendix #1, 6.1) . . . . .	6
4.2 Prototype #2 (Appendix #1, 6.2) . . . . .	6
4.3 Prototype #3 (Appendix #1, 6.3) . . . . .	6
4.4 Prototype #4 (Appendix #1, 6.4) . . . . .	6
4.5 Prototype #5 (Appendix #1, 6.5) . . . . .	6
<b>5 Phase #4: Team discussions and walkthrough</b>	<b>7</b>
5.1 Task #1 - Booking a new patient . . . . .	7
5.1.1 Task Walkthrough . . . . .	7
5.1.2 Summary . . . . .	9
5.2 Task #2 - Booking a no-show patient . . . . .	10
5.2.1 Task Walkthrough . . . . .	10
5.2.2 Summary . . . . .	12
5.3 Task #3 - Rescheduling an appointment . . . . .	13
5.3.1 Task Walkthrough . . . . .	13
5.3.2 Summary . . . . .	15
5.4 Task #4 - Marking a patient as a no-show . . . . .	16
5.4.1 Task Walkthrough . . . . .	16
5.4.2 Summary . . . . .	19
5.5 Task #5 - Updating a patient file . . . . .	19
5.5.1 Task Walkthrough . . . . .	19
5.5.2 Summary . . . . .	20
<b>6 Appendix #1 - Early Prototypes</b>	<b>21</b>
6.1 Prototype #1 . . . . .	21
6.2 Prototype #2 . . . . .	22
6.3 Prototype #3 . . . . .	23
6.4 Prototype #4 . . . . .	24
6.5 Prototype #5 . . . . .	26
<b>7 Appendix #2 - Final Prototype</b>	<b>29</b>
<b>8 Grading Sheet</b>	<b>33</b>

# 1 Phase #0: Setting the stage

In Phase 0 we explore the type of environment where a system can be deployed. This involves looking at how the particular environment operates with its users in its current form without any external influence or changes. We seek to quantify the details for them to use later down the road map where we start designing and developing the system.

## 1.1 Background Environment

Our project is concerned with helping front-line staff in medical clinics, laboratories and similar places to book patients with particular doctors. For this case and line of work, the majority of our users are already using a computer system to deal with most of their operational tasks. However, the existing software system design and usability have significant room for improvement regarding the booking of appointments, particularly with available information, efficiency, completeness of the workflow (detailed steps involved to book a single patient) and additional details that can benefit the entire process.

## 1.2 What the system will be used for as well as the general expectations that the system should satisfy

The system will primarily be used for booking patients for particular doctors in medical clinics. The general expectations the system should be able to satisfy are considered as follows:

- Allow users to obtain accurate patient details.
- Ask the user for all relevant required information to successfully book a patient.
- Allow the user to book patients strictly according to task descriptions, meaning no invalid step shall be performed by the user that can distract the user and prevent the user to complete the main goal (complete booking).
- Provide users complete control of the information that can be created, updated and deleted as per the user's wish.
- Allow users to keep track of patients who do not show up with a no-show fee that is tagged onto a booked patient's profile.

## 1.3 System Constraints

There are perhaps many systems constraints of any given system in general, and often few of them can be an obstacle in reaching the desired goal.

- **Budget:** Lack of budget can prevent us from expanding the design of the system to be able to serve all number of tasks as described in this document. For example, lack of computer power, internet speed and the inconsistent power supply may result in unwanted disruptions in the system and result in failed bookings. The scope of the design is adversely affected.
- **Inadequate training:** If the current users lack basic technical skills, for example how to operate a modern computer this can lead to inefficient use of the system regardless of an ideal design. The system relies on a modern computer and is run on a web server and therefore involves the application of basic technical knowledge.
- **Operating System:** Older versions of an operating system lacking the most recent network features may hold back an ideal design of the system. To elaborate the latest network protocols can handle web applications with better security and efficiency leading to fast load times and usability.
- **Security:** The front-line staff or clinic receptionists that are mostly considered to be primary users of the system may not always possess a strong technical background. At times patients' confidential information can be leaked regardless of adequate security measures in the system design. Problems such as this, arising from human behaviour are difficult to deal with from a design standpoint.

## 2 Phase #1: Identification

Phase 1 provides background and directly approaches the task descriptions where the users are expected to share details of their daily work that may or may not involve any particular system. The objective is to obtain key details concerning the user's tasks, goals and work as accurately as possible that can be translated to the system elements or components as we continue developing it.

### 2.1 Expected Types of Users

The types of users in our project are very limited. There are also some general assumptions we can make, our users will get by with limited technical knowledge as long as they are adept at using computers/mobile phones. Front-line staff are our users, particularly medical clinic receptionists, nurses, office staff, laboratory staff and so on. They share similar backgrounds most of the time.

### 2.2 Work Contexts

A typical scenario is as follows for our users: Let us name our user, Michelle. Michelle would start her day as usual in a front desk environment with standard computer equipment, desks, table and ideal lighting. She works at a medical clinic for this example and would wait for people (patients) to come in with their inquiries. She would talk with the patients and if the patient decides to see a doctor at this particular clinic Michelle would start preparing the system to be able to book the patient in. She then navigates to the patient management page and acquires information from the patient and creates a profile on them, after which she is able to book them in. Michelle is required to be sitting at the front desk for all operations and the complete process. It is important for Michelle to obtain the correct details about the patient.

### 2.3 Concrete Task Examples

To gather the following task examples, our group travelled to two walk-in medical clinics in Calgary to interview the front-desk receptionists directly about their typical workflow with regard to managing patient requests for appointments. Front desk receptionists at clinics represent a typical end-user of our system. The personas "Sarah" and "Emma" represent two of the receptionists who we interviewed. Task descriptions were generated based on the notes and recordings of our verbal interviews.

- Sarah works at the medical clinic and is a receptionist at the front desk working with prospective patients directly. When a patient walks in, Sarah asks for the following information: the patient's phone number, birth date, and full legal name. After obtaining them from the patient she uses all the collected patient details to book an appointment at a future scheduled time in 10–15-minute slots in the calendar as that can be extended depending on each patient's request. Sarah chooses a doctor who is available or assigned to the patient and then books the patient successfully under the doctor. The patient details can be obtained in two minutes for future requests or appointments
- Emma deals with individuals who have not attended their appointments in the past and want to schedule/reschedule an appointment. In this scenario, the patient walks into the clinic. Emma then asks for the patient's phone number to pull up their file. After verifying she has the correct file and that the patient indeed has not shown up in the past, she informs them that they must pay a no-show fee of \$30 before she is able to reschedule them. Once Emma has collected the no-show payment from the patient, Emma is able to reschedule their appointment. She will provide the available times of the available doctors to the patient. The patient has chosen a time, and Emma schedules them with the chosen doctor.
- Sarah is approached by a patient who already has a booking. The patient requests Sarah to reschedule their existing appointment. Sarah takes the patient's information and pulls up their file. Sarah clicks on the existing appointment and clicks on "modify appointment". Sarah proceeds to tell the availability of their doctor to the patient. Sarah re-books the patient with their new selected time and date with their doctor.

- Sarah checks her calendar and notices a patient has not shown up for their appointment. Sarah waits 15 minutes before opening the patient's file to attach a no-show tag to their file. She marks the patient as a no-show, resulting in the file being marked as a no-show.
- Sarah is approached by a patient who would like to update information on their file. Sarah asks for the patient's health care and pulls up their file. The patient proceeds to tell Sarah what information they would like Sarah to update, and Sarah updates that information on their file.

### 3 Phase #2: Tentative list of requirements

#### 3.1 Calcare Requirements

Must Include	Should Include	May Include	Exclude
Appointment Bookings	Appointment Reminders	Desk Worker ID	Self Bookings
No-Show fees	Health Care Number	Optional Patient Notes	Appointment Room Number
Patient Information	Calendar View		
Previous Medical Records			

##### 3.1.1 Must Include:

- **Appointment Bookings:** As our product is an appointment scheduling app for healthcare professionals, it is absolutely essential that our application includes this feature and that is why it is placed in the must-include category.
- **No-Show Fees:** A fee that is given to patients that don't show up to their appointments and in order to book a future appointment for the patient, any previous outstanding no-show fees must be paid. Since this directly impacts the bookings by either not showing up and paying the fee or paying the fee to book future appointments, it is a must-include feature.
- **Patient Information:** This allows for faster booking times for returning patients and would contain personal emergency information such as a patient's birthday, phone number, address, emergency contacts, etc. Patient information is required at almost any healthcare facility a person can go to and that is why it is a must-include.
- **Previous Medical Records:** In order for doctors to be caught up to speed with new patients or to keep up to date with current patients, they look at previous medical records including blood test reports, x-ray results, previous injuries, etc. Since this information is crucial for doctors to know about their patients, it has been placed in the must-include section.

##### 3.1.2 Should Include:

- **Appointment Reminders:** These would be sent out through either text or email and would be helpful for both the patient and the staff as it would help reduce the number of no-shows. It is not mandatory that the patient be reminded but since it would be beneficial for both the patient and the staff, we have placed it in the should include section.
- **Health Care Number:** Used to get patient information and for payments. However, in emergency situations, a patient can be taken in without a health care card and can provide it later if they want their provincial health care insurance plan to cover the charges. Seeing as how it is required for the most part but can be overlooked if needed, it is placed in the should include category.
- This is a viewing feature used to help sort, manage, and adjust patient appointment bookings. It is not necessary for this to be implemented as there are other design implementations that can accomplish the same task, but a calendar view is the most efficient and simple. Given that it is not the only option, but is still a very good option, it belongs in the should include category.

##### 3.1.3 May Include:

- **Desk Worker ID:** In order to log into the system the receptionist's ID will be required and in case of human error, it would be helpful to know which receptionist was in charge of the booking as there are typically many receptionists on site at once in most medical facilities. Given that not including this feature would not affect the overall functionality much or at all even, we have placed it in the may include the category.

- **Optional Patient Notes:** The patient may have extra details they would like their healthcare professional to know about prior to the appointment. Since the notes are optional and some patients may not have anything they would like to outline in advance for their doctor, this has been deemed a may-include feature.

### 3.1.4 Exclude:

- **Self Bookings:** Our product is easy to use but it may still be difficult for some patients to self-book based on their comfort level with technology, access to technology that would support this software, and access to a stable network. Also, given that we would have professional healthcare staff using this software already for appointments made in person or on the phone, it could get complicated and overbookings could occur. To avoid these issues altogether we placed self-bookings in the exclude section.
- **Appointment Room Number:** There is little to no use in knowing the room number beforehand as it does not affect the patient or the doctor in any way. Most of the time the patient is guided to the room from the waiting room by an appointed healthcare staff anyway, and for these reasons, the appointment room number is also in the excluded section.

## 4 Phase #3: Prototyping

### 4.1 Prototype #1 (Appendix #1, 6.1)

This mock-up was for the actual booking screen and the vision was a straightforward way for doctors to be booked individually based on their availability on a chosen date. As shown in the mock-up image, filtering doctors by availability on a given day would help narrow down the list but would be problematic if there are no time slots that work on that day. In that case, the receptionist would have to back out and then select the next doctor to see if the timing for them works and so on.

Additionally, a little note was made at the bottom which exemplified the idea of a staff schedule being implemented. In the staff schedule page mock-up, each of the doctor's availability for a given week is provided, and the ability to scroll through different weeks is provided at the top right. No changes were made to this page as everything incorporated into the design seemed necessary and minimal so there was not anything to add or take away from the original mock-up.

### 4.2 Prototype #2 (Appendix #1, 6.2)

This mock-up was particularly for the booking page where the user scans over three actions on one page and chooses them accordingly to finalize a booking for the patient. The vertical boxed list on the left would only have available doctors automatically filtered out on a given day and time that is shown at the bottom. The time at the bottom can be changed and an updated list of doctors will be shown. The patient information remains static on the right center of the page.

For the final design, the patient information section has been incorporated. Rest has not been taken into consideration since it can be confusing for the user to deal with these components together.

### 4.3 Prototype #3 (Appendix #1, 6.3)

The idea of this prototype was to make it easy for users to simultaneously view and schedule bookings so that they don't have to constantly switch between an appointments booking page and an appointments viewing page. A calendar view seemed to be the simplest way for users to quickly gain a visual understanding of the overall status of the clinic. Key features of this prototype include color coding for different appointment types, appointment filtering by a doctor, and the ability to schedule appointments via a pop-up window that appears when an appointment time slot is clicked on the screen.

An element of this prototype that was abandoned in the final prototype is the "Search patient" bar in the appointment scheduling popup, as we have opted to select the desired patient prior to booking them. This makes it simpler to ensure the correct patient was selected, as more patient information can be displayed in a dedicated patient search than in a drop-down menu.

### 4.4 Prototype #4 (Appendix #1, 6.4)

This mock-up was used to make the booking process easier. The idea was to create boxes which demonstrate which doctor is working for the particular week selected. Once the user selects a doctor the user can then select the patient assigned to the doctor. Once the user clicks on the patients they can then view the patient's information such as birthday, phone number, address, etc. On the left side, the user can make notes or view notes made regarding the patient or the patient's current medical issues. Under the patient's name, the user can set a reminder option for the patient to be reminded regarding their appointment.

### 4.5 Prototype #5 (Appendix #1, 6.5)

The idea of this prototype was to explain in detail how the booking process works. Once the user selects a certain doctor, they then select the patient. Once that's done the user is directed to a page with the patient's information such as the patient's last visit, any outstanding balance, contact information, address, birthday and most importantly the status of the appointment. The status of the appointment will define whether the patient receives a "no-show" tag or not. When the user clicks on "Book Appointment", they then are directed to a calendar page to select a time slot. After the time slot has been selected, the appointment has been booked. The user has an option to set a reminder to remind the patient regarding their appointment closer to the date.

## 5 Phase #4: Team discussions and walkthrough

### 5.1 Task #1 - Booking a new patient

#### 5.1.1 Task Walkthrough

Step Number	Step Description	Knowledgeable?	Motivated	Problems & Solutions
1	User clicks the "patient page" icon,	Yes, all actions related to patients are intuitively accessed by the "patient page" button.	Yes, the user needs to go to view to the patient page in order to book appointments for any patient.	N/A
2	User clicks "New Patient" on the Quick Options menu.	Yes, all fields on the Quick Options menu are explicitly labelled and "New Patient" is right at the top.	Yes, the user needs to create a file for the patient in order to book their appointment.	N/A
3	User enters patient information gathered from the patient.	Yes, all information fields are clearly labelled and appear on the same screen.	Yes, the user needs to enter all patient fields in order to proceed.	<ul style="list-style-type: none"> <li>• Problem #1: The user could enter patient information wrong.</li> <li>• Solution #1: Add a verification page to verify details before submitting.</li> </ul>
4	User validates that there are no error messages being displayed.	Yes, the errors are clearly displayed and explain the issue.	Yes, the user needs to ensure the information entered is correct before being able to book an appointment.	N/A
5	User clicks the "Create & Book" Button.	Yes, the button is clearly labelled and matches the user's current goal of booking a new patient's appointment.	Yes, the user's goal is to both create a new patient file and book them in.	N/A

6	User selects desired appointment time from the Calendar Page	User has asked the patient for their preferred appointment times and preferred doctor.	Yes, the user's goal is to complete the patient's booking.	<ul style="list-style-type: none"> <li>• Problem #1: It may be unclear to the user that they have to click on a date to trigger the appointment booking pop-up window.</li> <li>• Solution #1: Could noticeably change the cursor when the user hovers over the calendar, or could highlight blocks of time in the calendar when they hover over.</li> </ul>
7	User selects Doctor, Appointment Type and Time for an appointment.	Yes, fields are clearly labelled and the user has verbally requested the required information from the patient.	Yes, the user's goal is to schedule an appointment for the given patient.	<ul style="list-style-type: none"> <li>• Problem #1: User may have selected the incorrect date when clicking the calendar view.</li> <li>• Solution #1A: Display the date of the appointment on the "Schedule appointment" pop-up window.</li> <li>• Solution #1B: Make appointment date an editable field in the "schedule appointment" window or make it possible to click &amp; drag to move appointment blocks.</li> </ul>
8	User clicks on the "Book Appointment" button.	Yes, the button is clearly labelled. If booking an appointment is a success, a success message is displayed.	Yes, the user's goal is to submit the fields they have just entered.	N/A

### 5.1.2 Summary

Pros	Cons
<ul style="list-style-type: none"><li>• Fast appointment creation thanks to shortcuts.</li><li>• Fields, pages and buttons are intuitively labelled.</li></ul>	<ul style="list-style-type: none"><li>• Potential for user error when entering information.</li><li>Lack of error-checking.</li><li>• May be unclear how to schedule appointments in the calendar view.</li></ul>

## 5.2 Task #2 - Booking a no-show patient

### 5.2.1 Task Walkthrough

Step Number	Step Description	Knowledgeable?	Motivated	Problems & Solutions
1	User clicks on the "patient page" icon within the taskbar.	Yes, all actions related to patients are intuitively accessed by the "patient page" button.	Yes, the user needs to go to view the patient page in order to book appointments for any patient.	N/A
2	User clicks "Patient Lookup" on the Quick Options menu.	Yes, all fields on the Quick Options menu are explicitly labelled.	Yes, the user needs to look up a file for the patient in order to clear their file of the no-show fee and book an appointment for them.	N/A
3	User enters the patient's healthcare number.	Yes, all fields on the "Patient Lookup" are explicitly labelled.	Yes, the user needs to enter this information to be able to locate the patient's file.	<ul style="list-style-type: none"> <li>• Problem #1: User incorrectly enters the patient's information.</li> <li>• Solution #1A: Using basic regular expressions, we can verify that the information meets the expected criteria.</li> <li>• Solution #1B: Add an extra verification page to verify the patient information is correct before moving on.</li> </ul>

4	User will click on the no-show tag to see the required payment.	Yes, the information on the page is explicitly marked and provides all the required details.	Yes, the user must view this information to further proceed with the booking process.	<ul style="list-style-type: none"> <li>• Problem #1: The no-show tag may not be noticed right away.</li> <li>• Solution #1A: Disable (gray out) the options to book patients and make the tag much more visible.</li> <li>• Solution #1B: Go to a modified patient page, that only contains their information, and shows the details for the no-show tag.</li> </ul>
5	User will now mark the no-show tag as paid, removing the tag from the file.	Yes, the button is explicitly labelled.	Yes, the user will be motivated to remove this tag so that they can book the patient.	N/A
6	User selects desired appointment time from the Calendar Page.	User has asked the patient for their preferred appointment times and preferred doctor.	Yes, the user's goal is to complete the patient's booking.	<ul style="list-style-type: none"> <li>• Problem #1: It may be unclear to the user that they have to click on a date to trigger the appointment booking pop-up window.</li> <li>• Solution #1: Could noticeably change the cursor when the user hovers over the calendar, or could highlight blocks of time in the calendar when they hover over.</li> </ul>

7	User selects Doctor, Appointment Type and Time for an appointment.	Yes, fields are clearly labelled and the user has verbally requested the required information from the patient.	Yes, the user's goal is to schedule an appointment for the given patient.	<ul style="list-style-type: none"> <li>• Problem #1: User may have selected the incorrect date when clicking the calendar view.</li> <li>• Solution #1A: Display the date of the appointment on the "Schedule appointment" pop-up window.</li> <li>• Solution #1B: Make appointment date an editable field in the "schedule appointment" window or make it possible to click &amp; drag to move appointment blocks.</li> </ul>
8	User clicks on the "Book Appointment" button.	Yes, the button is clearly labelled. If booking an appointment is a success, a success message is displayed.	Yes, the user's goal is to submit the fields they have just entered.	N/A

### 5.2.2 Summary

Pros	Cons
<ul style="list-style-type: none"> <li>• Simple and straightforward design, making the task easy to do.</li> <li>• Information is explicitly labelled, to avoid any confusion.</li> </ul>	<ul style="list-style-type: none"> <li>• No-show tag may not be clear.</li> </ul>

### 5.3 Task #3 - Rescheduling an appointment

#### 5.3.1 Task Walkthrough

Step Number	Step Description	Knowledgeable?	Motivated	Problems & Solutions
1	User clicks on the "patient page" icon within the taskbar.	Yes, all actions related to patients are intuitively accessed by the "patient page" button.	Yes, the user needs to go to view the patient page in order to book appointments for any patient.	N/A
2	User clicks "Patient Lookup" on the Quick Options menu.	Yes, all fields on the Quick Options menu are explicitly labelled.	Yes, the user needs to look up a file for the patient in order to clear their file of the no-show fee and book an appointment for them.	N/A
3	User enters the patient's healthcare number.	Yes, all fields on the "Patient Lookup" are explicitly labelled.	Yes, the user needs to enter this information to be able to locate the patient's file.	<ul style="list-style-type: none"> <li>• Problem #1: User incorrectly enters the patient's information.</li> <li>• Solution #1A: Using basic regular expressions, we can verify that the information meets the expected criteria.</li> <li>• Solution #1B: Add an extra verification page to verify the patient information is correct before moving on.</li> </ul>
4	User selects "modify appointment" which redirects them to a calendar page that outlines where their current appointment is.	Yes, all actions related to choosing to modify a patient's appointment are explicitly labelled.	Yes, the user needs to go to the calendar page in order to select an available time for the patient to reschedule to.	N/A

6	User selects desired appointment time from the Calendar Page	User has asked the patient for their preferred appointment times and preferred doctor.	Yes, the user's goal is to complete the patient's booking.	<ul style="list-style-type: none"> <li>• Problem #1: It may be unclear to the user that they have to click on a date to trigger the appointment booking pop-up window.</li> <li>• Solution #1: Could noticeably change the cursor when the user hovers over the calendar, or could highlight blocks of time in the calendar when they hover over.</li> </ul>
7	User selects Doctor, Appointment Type and Time for an appointment.	Yes, fields are clearly labelled and the user has verbally requested the required information from the patient.	Yes, the user's goal is to schedule an appointment for the given patient.	<ul style="list-style-type: none"> <li>• Problem #1: User may have selected the incorrect date when clicking the calendar view.</li> <li>• Solution #1A: Display the date of the appointment on the "Schedule appointment" pop-up window.</li> <li>• Solution #1B: Make appointment date an editable field in the "schedule appointment" window or make it possible to click &amp; drag to move appointment blocks.</li> </ul>
8	User clicks on the "Book Appointment" button.	Yes, the button is clearly labelled. If booking an appointment is a success, a success message is displayed.	Yes, the user's goal is to submit the fields they have just entered.	N/A

### 5.3.2 Summary

Pros	Cons
<ul style="list-style-type: none"><li>Fields, pages and buttons are intuitively labelled.</li></ul>	<ul style="list-style-type: none"><li>May be unclear how to schedule appointments in the calendar view.</li></ul>

## 5.4 Task #4 - Marking a patient as a no-show

### 5.4.1 Task Walkthrough

Step Number	Step Description	Knowledgeable?	Motivated	Problems & Solutions
1	User checks the calendar view.	Yes, a calendar is an intuitive way for the user to view the status of current appointments.	Yes, checking in patients for appointments is part of the user's job.	<ul style="list-style-type: none"> <li>• Problem #1: User may not realize that an appointment has lapsed without the patient showing up.</li> <li>• Solution #1: Add a "check-in" feature to indicate when patients show up for their appointments. Send users an alert or notification when patients have not been "checked in" for their appointments after the appointment has passed.</li> </ul>

2	User click on the appointment in the calendar view to bring up appointment details.	Yes, the calendar view is the main way that information about appointments is viewed.	Maybe. In the case of multiple receptionists, the user may not even be aware that this appointment was missed.	<ul style="list-style-type: none"> <li>• Problem #1: The receptionist might not know that a patient didn't show up, the application currently only supports booking and not checking in.</li> <li>• Solution #1: Implement a check-in system for appointments.</li> <li>• Problem #2: User has to mentally calculate how much time has passed since the missed appointment was supposed to occur.</li> <li>• Solution #2: Could grey out or change the color of appointments in the calendar for which a given amount of time has passed without the patient being checked in.</li> </ul>
3	User navigates to patient look-up to look up the patient who missed their appointment.	Yes, this should take at most two clicks and the taskbar is clearly labelled.	Yes, the user wants to add a no-show tag to the given patient's file.	N/A

4	User looks up a patient by name.	Maybe, the user might not be able to mentally remember the patient information that they saw in the calendar view.	Yes, the user wants to find the given patient to add a no-show tag to their file.	<ul style="list-style-type: none"> <li>• Problem #1: User has to mentally recall patient information when looking it up.</li> <li>• Solution #1A: Could make it possible to click on the patient's information when viewing the appointment in the calendar view (via hyperlink).</li> <li>• Solution #1B: Could simply automate the process of adding a no-show tag to patient files, if the check-in process mentioned earlier is implemented.</li> </ul>
5	User selects the desired patient from search results.	No, if there are multiple patients with the same name, the user probably doesn't have enough information to know which one is the patient they're looking for.	Yes, the user needs to look up the patient to access their file.	Same as above.
6	User clicks "update patient" on the patient's file.	Yes, if the user has made it here then they had the information required to find the patient. The update patient button is clearly labelled.	Yes, the user's goal is to update a patient's file.	N/A
7	User checks the "no-show" checkbox on the patient's file.	Yes, the no-show checkbox is a familiar field in every patient's file.	Yes, this step is the main goal of the user during this task.	N/A

#### 5.4.2 Summary

Pros	Cons
<ul style="list-style-type: none"> <li>Calendar view is a good way to get a bird's eye view of upcoming/missed appointments.</li> </ul>	<ul style="list-style-type: none"> <li>User has to memorize too much information.</li> <li>User has to manually navigate between several windows to complete the task.</li> <li>Lack of a check-in system to even verify if patients have missed their appointment.</li> </ul>

### 5.5 Task #5 - Updating a patient file

#### 5.5.1 Task Walkthrough

Step Number	Step Description	Knowledgeable?	Motivated	Problems & Solutions
1	User clicks on the "patient page" icon within the taskbar.	Yes, all actions related to patients are intuitively accessed by the "patient page" button.	Yes, the user needs to go to view the patient page in order to book appointments for any patient.	N/A
2	User clicks "Patient Lookup" on the Quick Options menu.	Yes, all fields on the Quick Options menu are explicitly labelled.	Yes, the user needs to look up a file for the patient in order to clear their file of the no-show fee and book an appointment for them.	N/A
3	User enters the patient's healthcare number.	Yes, all fields on the "Patient Lookup" are explicitly labelled.	Yes, the user needs to enter this information to be able to locate the patient's file.	<ul style="list-style-type: none"> <li>Problem #1: User incorrectly enters the patient's information.</li> <li>Solution #1A: Using basic regular expressions, we can verify that the information meets the expected criteria.</li> <li>Solution #1B: Add an extra verification page to verify the patient information is correct before moving on.</li> </ul>

4	User proceeds to update the information that has been requested to be updated by clicking any fields that need to be updated.	Yes, all fields on the patient file are explicitly marked.	Yes, the user wants to enter the updated information as requested.	<ul style="list-style-type: none"> <li>• Problem #1: User incorrectly enters the patient's information.</li> <li>• Solution #1A: Using basic regular expressions, we can verify that the information meets the expected criteria.</li> <li>• Solution #1B: Add an extra verification page to verify the patient information is correct before moving on.</li> </ul>
5	User hits "Update Patient" to finalize the changes.	Yes, the button is explicitly labelled.	Yes, the user needs to confirm changes in order to finalize and update patient information.	N/A

### 5.5.2 Summary

Pros	Cons
<ul style="list-style-type: none"> <li>• Simple and straightforward designing, making the task easy to do.</li> <li>• Information is explicitly labelled, to avoid any confusion.</li> </ul>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>

## 6 Appendix #1 - Early Prototypes

### 6.1 Prototype #1

ref 10  
2  
B

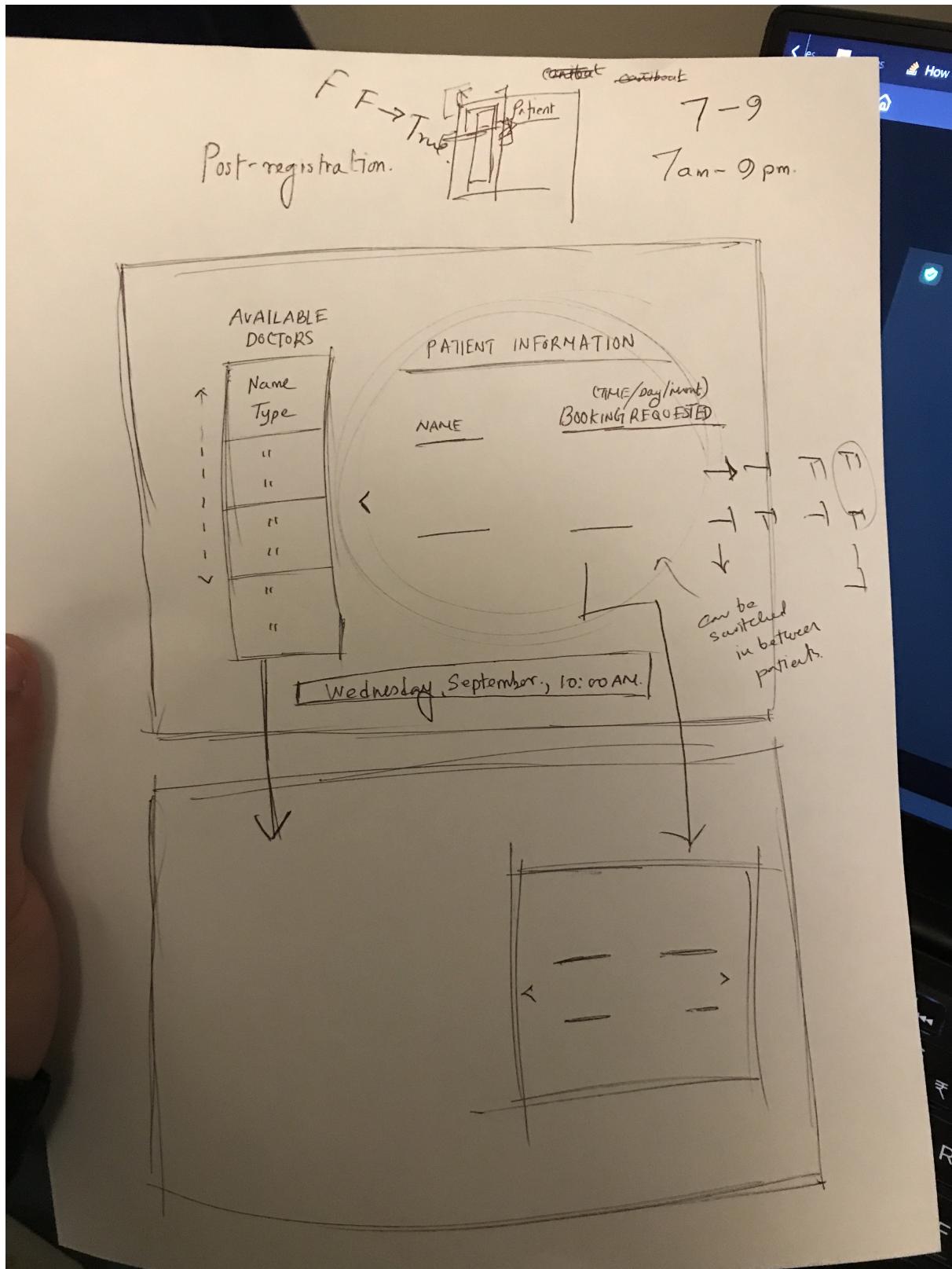
STAFF AVAILABILITY : * CURRENT DATE *		
	NAME AVAILABLE	<input type="button" value="Book / See Times"/>
	NAME AVAILABLE	<input type="button" value="Book / See Times"/>
	NAME UNAVAILABLE	<input type="button" value="Book / See Times"/>
	NAME AVAILABLE	<input type="button" value="Book / See Times"/>

\* We should also add a Staff Schedule page somewhere.

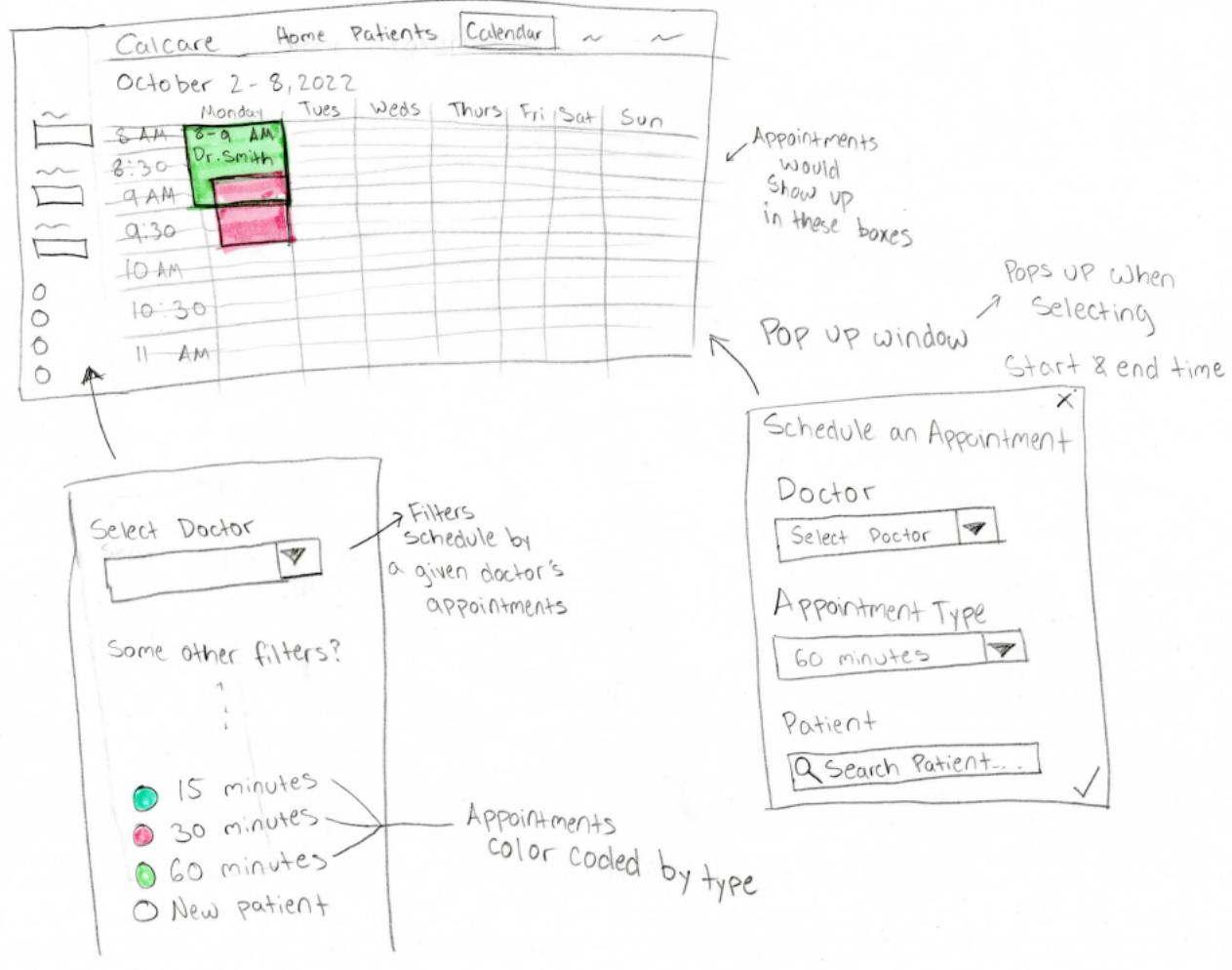
This Week    **SUN OCT 2<sup>nd</sup>, 2022 - SUN OCT 8<sup>th</sup>, 2022**

	2 SUNDAY	3 MONDAY	4 TUESDAY	5 WEDNESDAY	6 THURSDAY	7 FRIDAY	8 SATURDAY
DOCTOR #1 NAME ID # 1234567	Not Scheduled	8:00a-4:00p	Not Scheduled	...	...	...	...>
DOCTOR #2 NAME ID # 0000000	Not Scheduled	Not Scheduled	9:00a- 3:00p	...	...	...	...>
DOCTOR #3 NAME ID # 1111111	Not Scheduled	Not Scheduled	10:00a- 6:00p	...	...	...	...>
DOCTOR #4 NAME ID # 2222222	9:00a- 5:00p	Not Scheduled	Not Scheduled	...	...	...	...>
DOCTOR #5 NAME ID # 3333333	9:00a- 5:00p	10:00a- 6:00p	Not Scheduled	...	...	...	...>

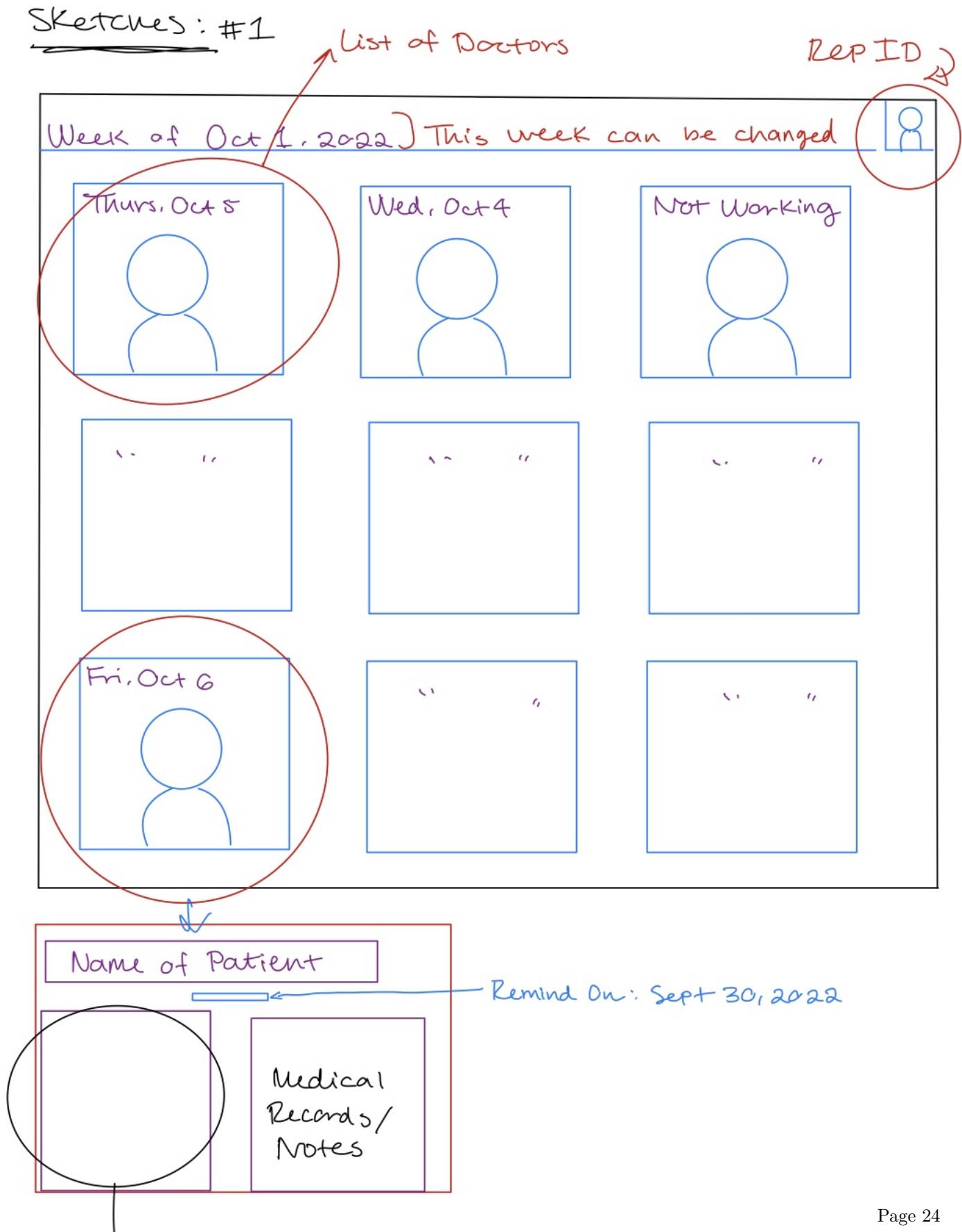
## 6.2 Prototype #2



### 6.3 Prototype #3

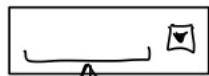


## 6.4 Prototype #4





Status:



Please select  
a option

Has a drop down menu  
which states "Armed" OR "No-Snow (\$30)"

Contact: 403-997-7735

Address: 555 Disney road NW

Birthday: Jan/01/2001

Health Card Number: ##

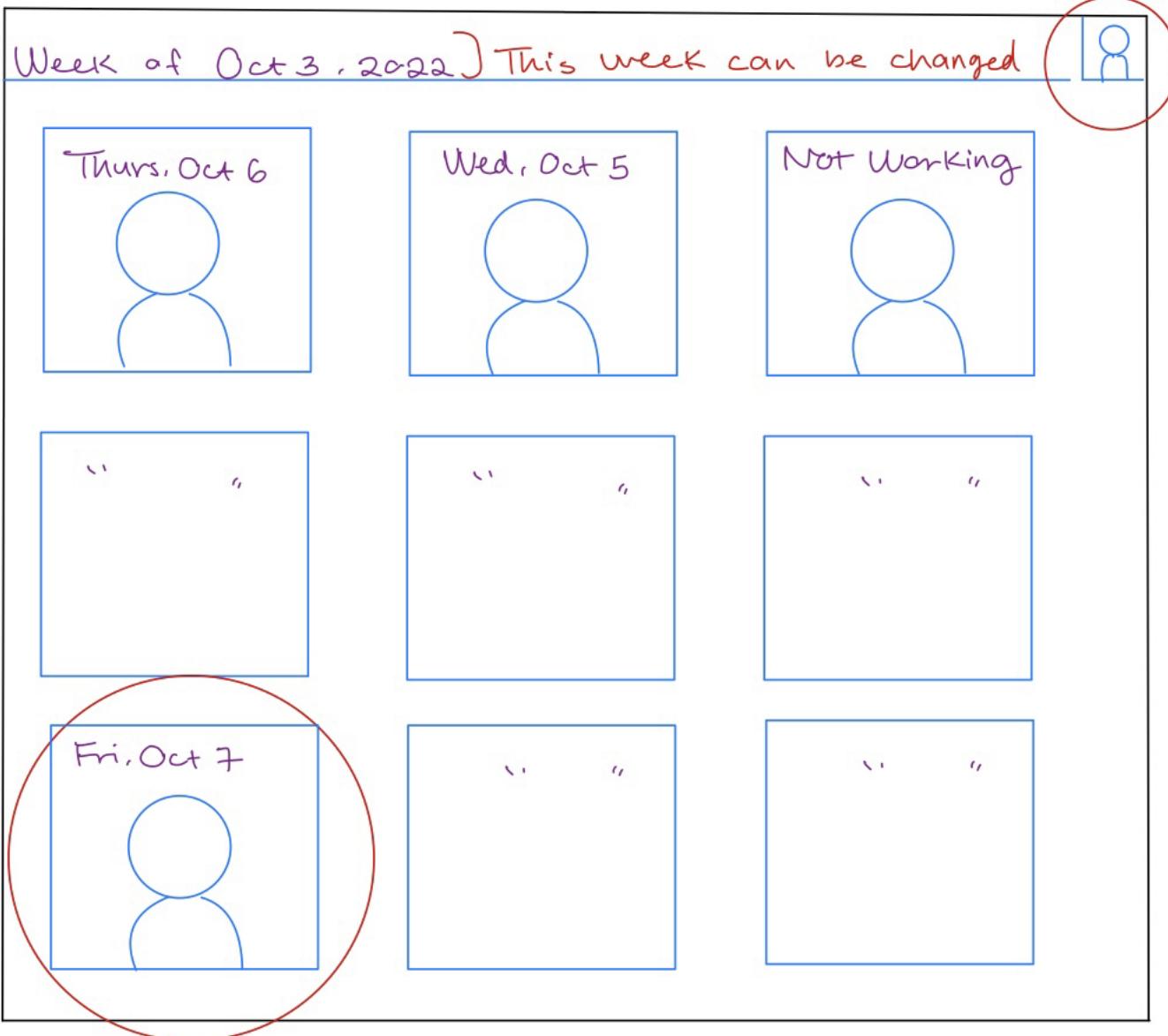
What was my idea?

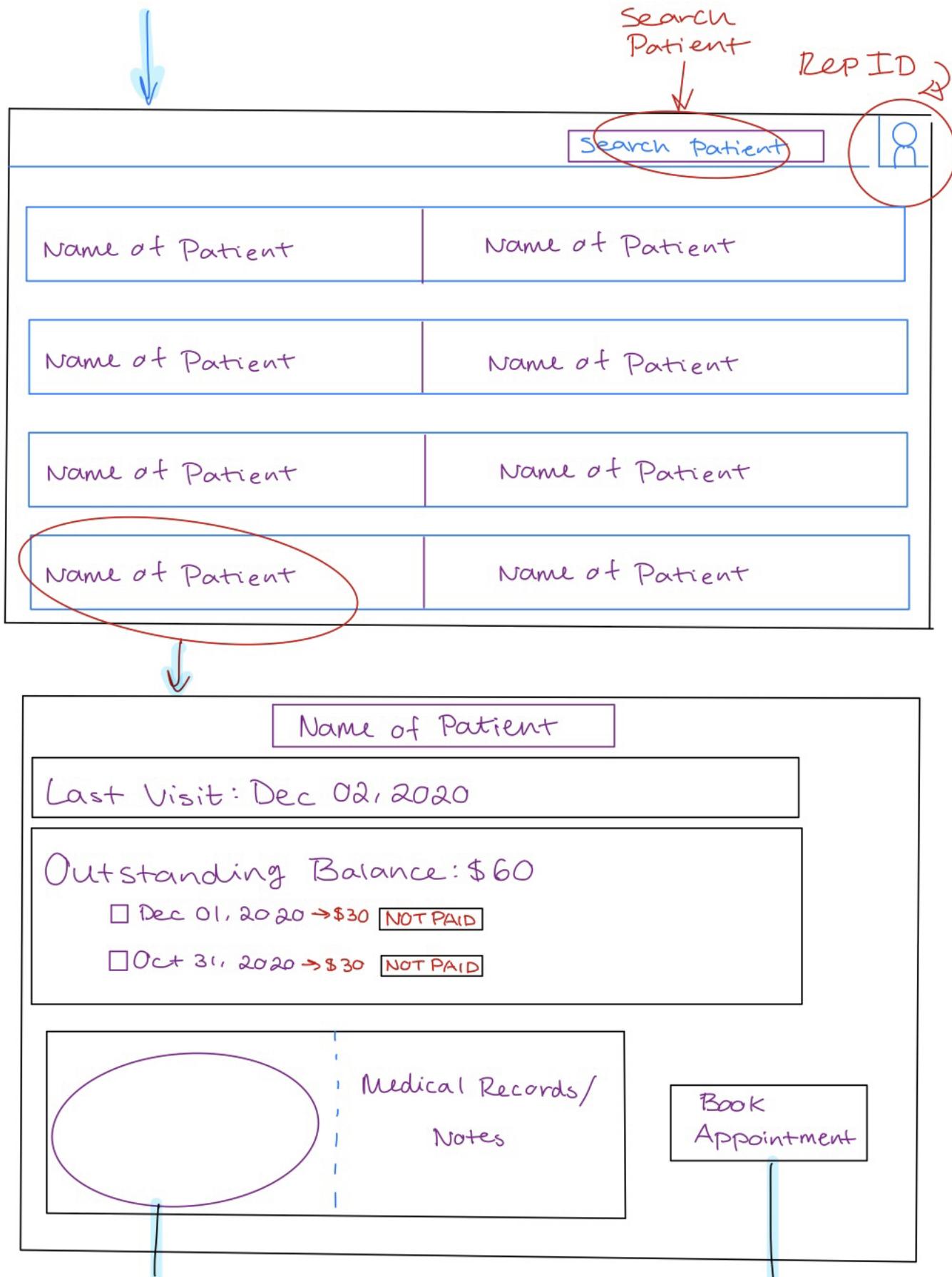
- ↳ My idea is to create boxes which demonstrate when the doctor is working next during that week.
- ↳ Once rep chooses a doctor we then can view the doctors patients.
- ↳ When you click on the patient you can see their personal info such as: B-day, phone #, address, etc.
- ↳ On the left side you can see or make notes of the patients previous or current medical issues.
- ↳ Under the patients name you can set a reminder option for the patient to be reminded for their appointment. This reminder text will be sent automatically.

## 6.5 Prototype #5

Sketches: #3

REP ID ↗





Status:    
 Please select a option

Has a drop down menu which states "Arrived" OR "No-show (\$30)"

Photo of Dr. ↗

Contact: 403-997-7735

Address: 555 Disney road NW

Birthday: Jan 01 / 2001

Health Card Number: ##

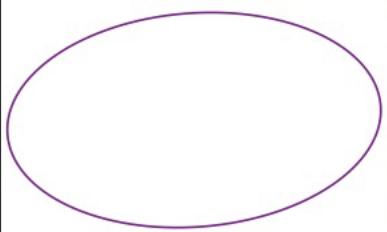
Time	Sunday	Monday	Tuesday
11:00am	-Emily Joe -Sam Joe		
11:30am			
12:00pm			
12:30pm	-Sally Ju -Emma Ju -Mia Ju -Bob Ju		
1:00pm			
1:30pm			
:			

Appointment Has  
Been Booked  
Successfully!

appointment will be  
booked during this time

Display Screen ↗

Name of Patient	Remind On: Oct 7, 2022
Last Visit: Dec 02, 2020	
Outstanding Balance: \$60 <input type="checkbox"/> Dec 01, 2020 → \$30 NOT PAID <input type="checkbox"/> Oct 31, 2020 → \$30 NOT PAID	

	Medical Records/ Notes	Appointment Booked (Monday Oct 12, 2022 @ 11:00am)
---	---------------------------	---

## 7 Appendix #2 - Final Prototype

Main Page:

Logo	Page Name	Date & Time
 Home  Calendar  Staff Schedule  Patient  Logout	<p><b>Important Notices</b></p> <ol style="list-style-type: none"> <li>1.) Dr. Bob is leaving for vacation in 2 weeks!</li> <li>2.) " "</li> <li>3.) " "</li> <li>4.) " "</li> </ol> <p><b>Doctors</b></p> <p>Dr. Bob: Hours → 8-8 Available Appointment → 10 Booked → 2</p>	<p><b>Home</b></p> <p><b>October 5, 2022</b> <b>12:30pm</b></p> <p><b>Shortcut Key</b></p> <p>F2 → Select a doctor F3 → Book appointment F4 → Update Patient F5 → New Patient</p>

Account Page:

Logo	Page Name	Date & Time
 Home  Calendar  Staff Schedule  Patient  Logout	 <p>Name: <input type="text"/></p> <p>Clinic: <input type="text"/></p> <p>Email: <input type="text"/></p> <p>Password: <input type="text"/></p> <p><b>Account Setting</b></p> <p><b>October 5, 2022</b> <b>12:30pm</b></p> <p><b>Confirm</b></p>	

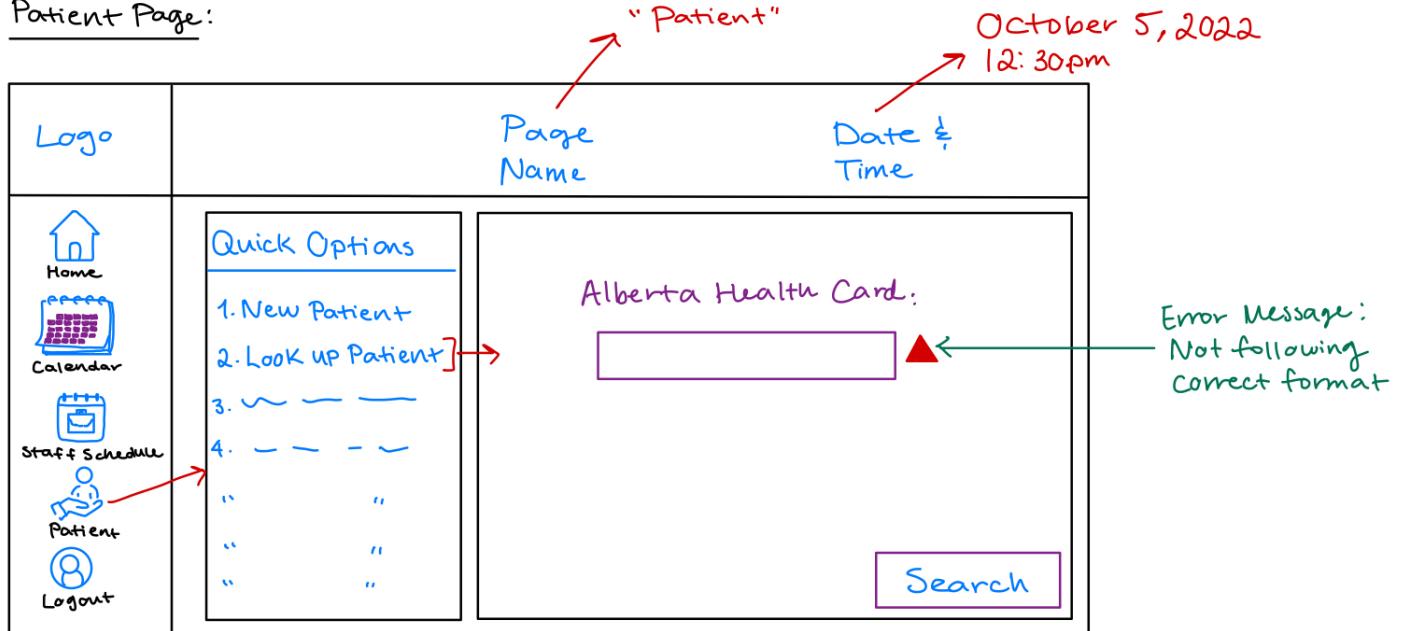
## Patient Page:

Logo	Page Name	Date & Time
 Home  Calendar  Staff & Schedule  Patient  Logout	<h3>Quick Options</h3> <ol style="list-style-type: none"><li>1. New Patient</li><li>2. Look up Patient</li><li>3. — — —</li><li>4. — — —</li><li>" " "</li><li>" " "</li><li>" " "</li></ol>	12:30pm

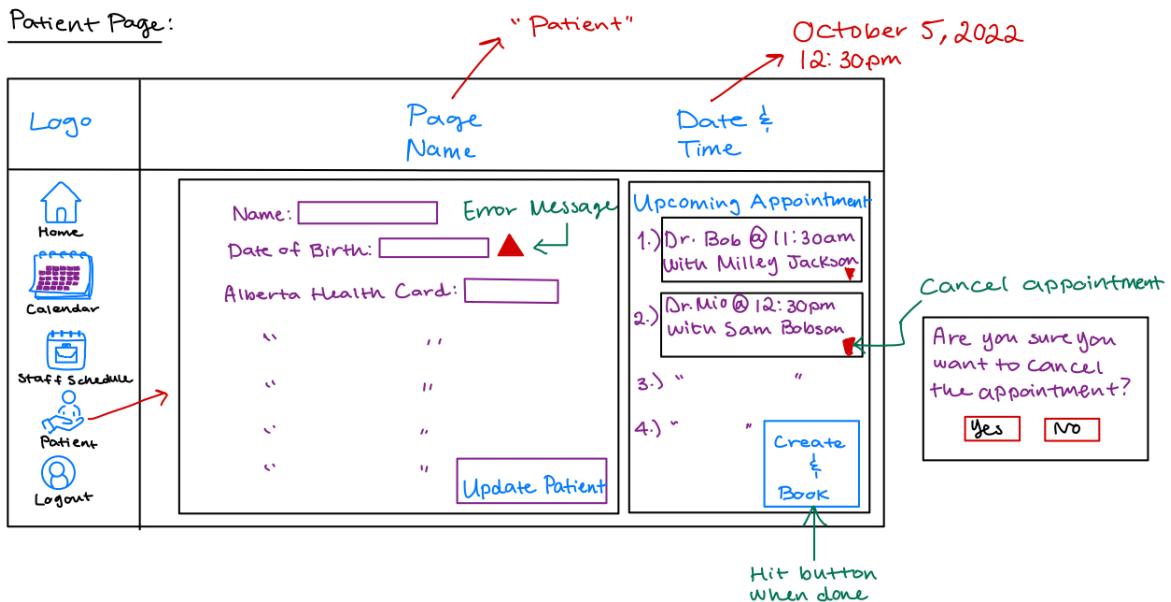
## Patient Page:

Patient Page:	
Logo	Page Name Patient
	Date & Time October 5, 2022 12:30pm
 Home  Calendar  Staff Schedule  Patient  Logout	<div style="display: flex; justify-content: space-between;"> <div style="width: 30%;"> <p><u>Quick Options</u></p> <ol style="list-style-type: none"> <li>1. New Patient</li> <li>2. Look Up Patient</li> <li>3. " "</li> <li>4. " "</li> <li>" "</li> <li>" "</li> <li>" "</li> <li>" "</li> </ol> </div> <div style="width: 40%; position: relative;"> <p>Name: <input type="text"/></p> <p>Date of Birth: <input type="text"/> <span style="position: absolute; top: -10px; left: 0; color: red;">▲</span></p> <p>Alberta Health Card: <input type="text"/></p> <p>" "</p> <p>" "</p> <p>" "</p> <p>" "</p> </div> <div style="width: 30%; border: 1px solid blue; padding: 5px;"> <p><u>Create File</u></p> <p><u>Create &amp; Book</u></p> <p>" "</p> <p>" "</p> </div> </div> <div style="margin-top: 10px; border: 1px solid green; padding: 5px; width: fit-content;"> <span style="color: green;">Error Message</span>  <span style="color: green;">Hit button when done</span> </div>

## Patient Page:



## Patient Page:



Calendar Page:

Logo	Page Name	Date & Time
 Home  Calendar  Staff Schedule  Patient  Logout	Oct 2 - Oct 8 <input type="button" value="&lt;"/> <input type="button" value="&gt;"/> Day Week Month Dr.Mio	October 5, 2022 12:30pm
	Time MON TUE WED THUR FRI SAT SUN 8:00AM Dr.Bob 8:30AM 9:00 AM 9:30AM Dr.Mio 10:00AM 10:30AM Dr.Si	Dr.Wu  Dr.Jay

POP-up Window

Schedule Appointment

Doctor

Select Doctor

Appointment Type

30 minutes

Time

9:00AM - 9:30AM

Appointment has  
 been booked  
 successfully!

## 8 Grading Sheet

### Grading Sheet

#### Structure and format

	<b>Included</b>	<b>Not Included</b>	
Portfolio in PDF	1	0	
Section separators	1	0	
Name on outside cover	1	0	
Name and contact information on the first page	1	0	
This grading sheet included in portfolio	4	0	
	<b>Complete</b>	<b>Missing portions</b>	<b>Not included</b>
Table of contents	2	1	0
	<b>Great: no problems</b>	<b>Good: a few minor problems</b>	<b>Poor: Problems throughout (your mark in other sections may also be affected as well)</b>
Appearance (organization, layout and whitespace)	6	4	0
	<b>No typos, grammatical or spelling errors, clear writing style</b>	<b>Minor typos or grammatical errors or spelling mistakes or some writing may be a bit vague</b>	<b>Problems in two areas (spelling, grammar, style)</b>
Language and writing style	7	5	3 0
	<b>Problems in all three areas</b>		

#### Setting the stage

**Clear and complete (yes)   Clear and complete (no)**

Background	1	0
------------	---	---

Expected uses of the system	1	0	
System constraints	1	0	
	<b>Lists user groups along with relevant skills and experience</b>	<b>Lists user groups with no additional information</b>	<b>Information not included</b>
Expected users	2	1	0
	<b>Clear &amp; complete</b>	<b>Some information missing or unclear</b>	<b>Information not included</b>
Work context	2	1	0
	<b>Spoke directly with actual users</b>	<b>Spoke with a representative of the user</b>	<b>Made it all up</b>
Approach for getting background information for tasks	2	1	0

Tasks

	<b>Appropriate No. (~5-7)</b>	<b>Fewer than what's needed for the usage of the system</b>	<b>No tasks were included in the portfolio</b>	
Number of tasks	2	1	0	
	<b>Covers all relevant activities</b>	<b>Missing a few important tasks</b>	<b>Missing many important tasks</b>	<b>No tasks were included in the portfolio</b>
Coverage of the tasks	8	6	2	0
	<b>No violations</b>	<b>A few minor violations</b>	<b>Many violations throughout</b>	<b>No tasks were included in the portfolio</b>
Do the tasks follow the properties of a good task?	8	6	2	0

Prototypes

	<b>Two or more</b>	<b>One</b>	
Number of versions/iterations	2	1	
	<b>Marked improvement from version to version</b>	<b>Few and/or superficial changes from version to version</b>	<b>No evolution between prototype versions</b>
Evolution of prototypes	6	2	0
	<b>Provides clear idea of how prototype changed from version to version</b>	<b>Describes changes but some parts are unclear</b>	<b>None</b>
Description of how prototypes evolved	4	2	0

Requirements

	<b>Requirements are grouped into categories with clear and detailed explanations based on the users and their tasks</b>	<b>Requirements are grouped into categories, no indication of how functions were put into particular categories</b>	<b>Requirements are shown in a single list, no attempt at prioritization</b>	<b>No requirements listed</b>
Description of system functions to be implemented	5	2	1	0

Walkthroughs

	<b>Walkthroughs for all relevant tasks</b>	<b>One</b>	<b>Zero</b>	
Number of walkthroughs performed	4	1	0	
	<b>Walkthroughs conducted, all or most usability problems were caught</b>	<b>Walkthroughs conducted, some minor problems were missed</b>	<b>Walkthroughs conducted, many minor or several serious problems were missed</b>	<b>Walkthrough not performed</b>

Results of conducting the walkthrough algorithm	10	8	4	0
	<b>Walkthrough results summarized for each scenario/task An analysis conducted that summarized for all tasks what are the high level and major problems</b>	<b>Walkthrough results summarized for each scenario/task but not for all tasks</b>	<b>Walkthroughs conducted and results shown in table but no additional analysis, summarizing problems</b>	
Analysis of walkthrough results	6	3	0	
	<b>Walkthroughs easy to follow (e.g., included diagrams at all relevant points of walkthrough, diagrams are annotated)</b>	<b>Some points of the walkthrough difficult to follow (e.g., walkthrough description didn't match interface, additional diagrams would have made things clearer)</b>	<b>Walkthroughs not conducted</b>	
Ease of following/tracing the walkthroughs	6	3	0	

Tutorial presentations

	<b>Provides clear background information, good tasks presented, requirements properly categorized</b>	<b>Minor problems: some background information unclear, minor violations in the descriptions of the tasks, requirements could better justified</b>	<b>Poor: task violate many properties of good tasks, or background missing or largely incomplete, requirements are not justified</b>	<b>No presentation</b>
First presentation:	4	3	1	0

Phase one and two

	<b>Walkthrough:</b> caught most problems, clear indication of what future improvements should be  <b>Prototype:</b> Gives a good feel for how the interaction unfolds, covers main system functions	<b>Walkthrough:</b> Missed a few minor problems in the walkthrough  <b>Prototype:</b> Some parts of the interaction unclear, a few minor system functions (relevant to task) or a major function is missing	<b>Walkthrough:</b> Missed many minor problems in the walkthrough or a few major usability problems  <b>Prototype:</b> several main system functions missing	<b>Walkthrough:</b> Many serious problems were missed in the walkthrough  <b>Prototype:</b> main system functions were missing
Second presentation: Phase three & four	4	3	1	0
All team members completed all weekly MVP surveys to portfolio due date	Complete	Incomplete		