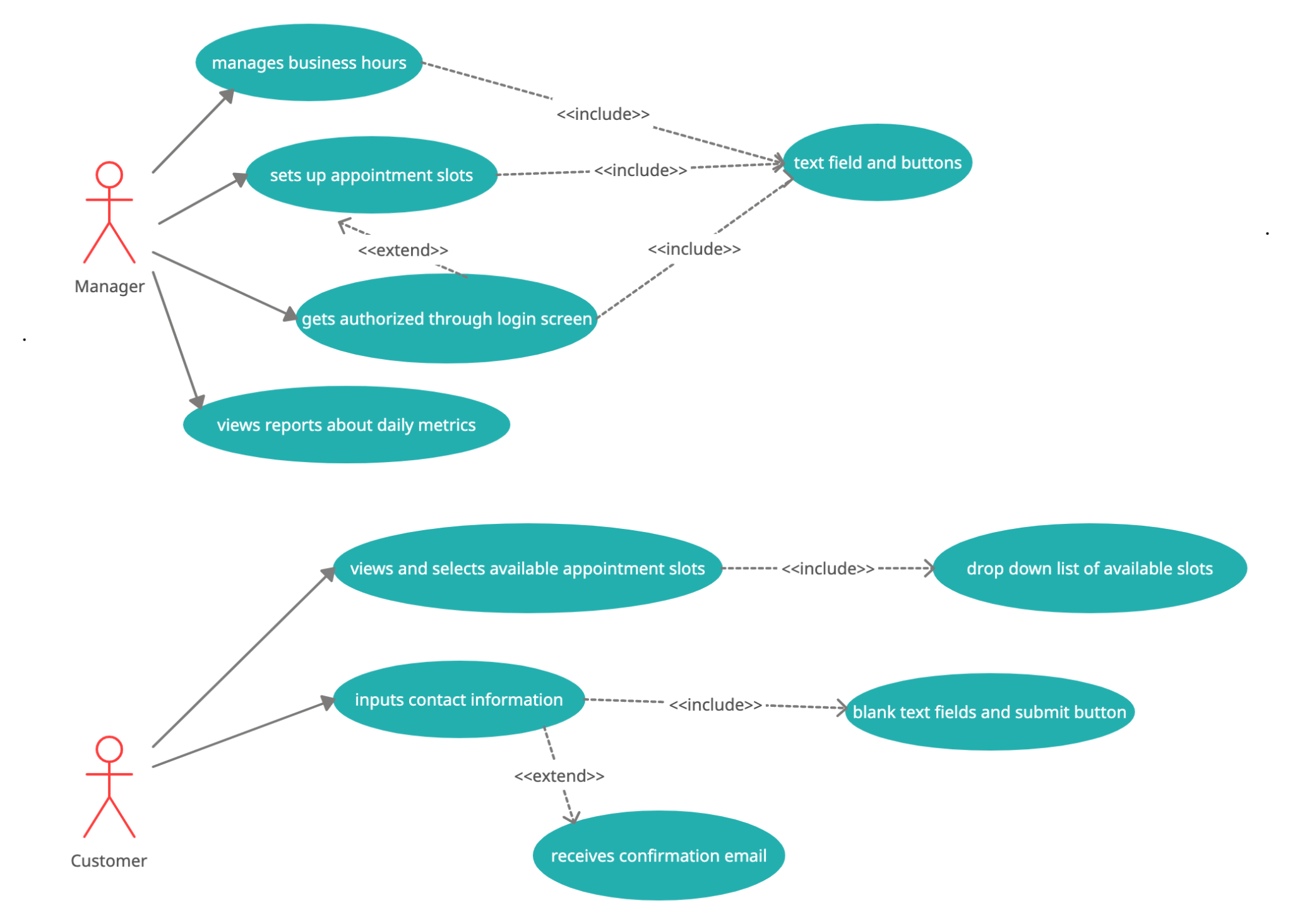
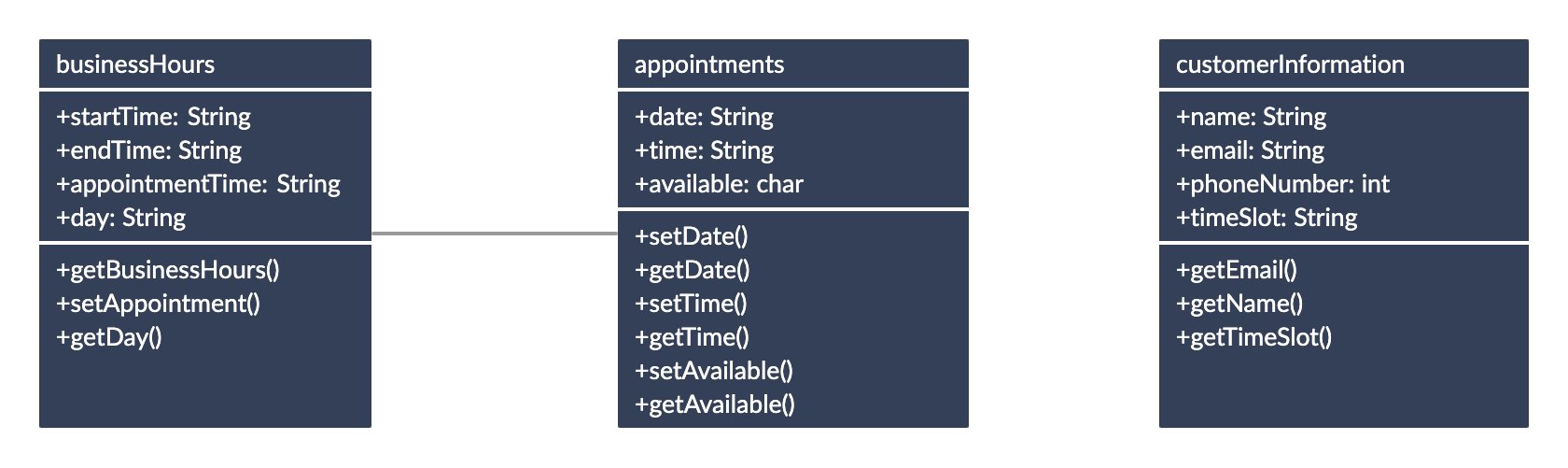
**Criterion B: Design**

**Use Case Diagrams:**

**Version 1:**

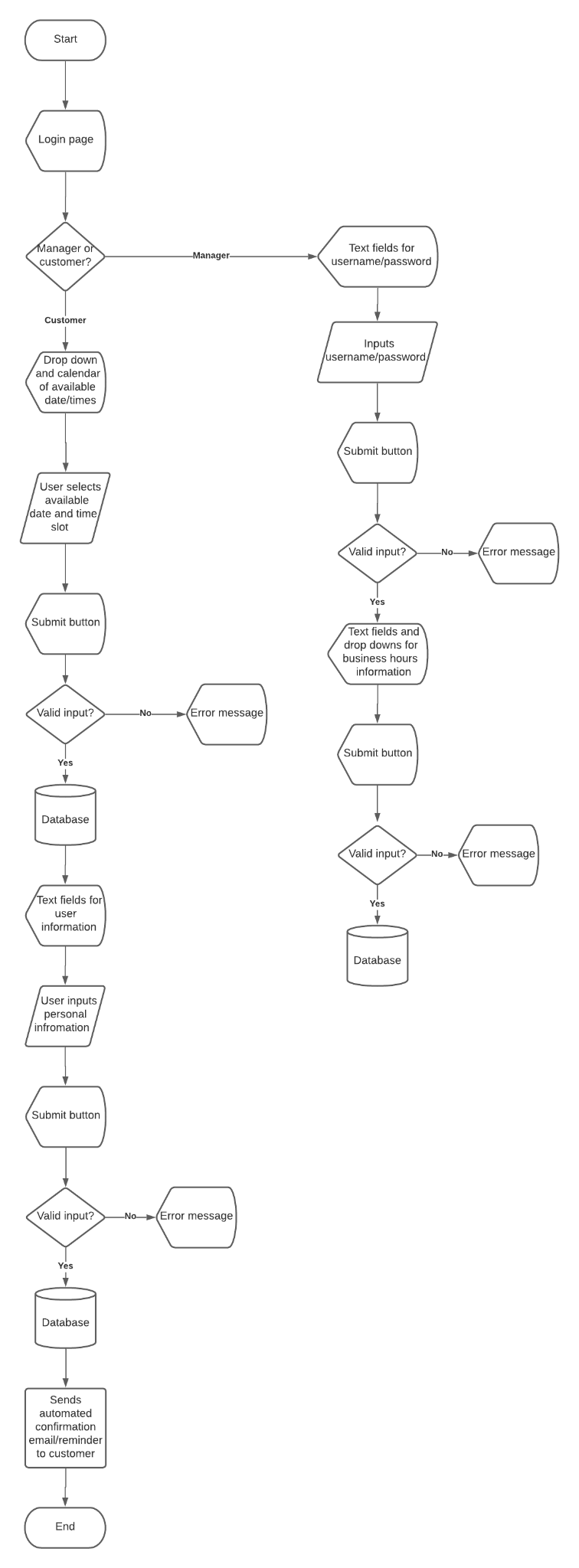


**Class Diagrams:**

**Version 1:**

**Flowcharts:**

**Version 1:**



**Algorithms:**

**Version 1:**

The algorithm which is described in detail below, is regarding displaying and processing the appointment dates/times for the testing site.

| **Step** | **Process** |
| --- | --- |
| 1 | Store the business hour information (start time, end time, days of week, time length of test) in a database table called “businessHours”. |
| 2 | Based on manager input (from database), mathematically calculate the amount of and timings of appointments for the days the testing site is open. |
| 3 | Insert calculated appointment dates/times (based on day) in a database table called “appointments”. |
| 4 | Initially mark all appointments as available in the database table column labeled as “availability” in the “appointments” table. |
| 5 | Display a calendar graphic for users to select available appointment dates. Color code the calendar to distinguish the dates that are available and not available (holidays) |
| 6 | Display a drop-down list of all available times from the database table called “appointments” with “availability” column values marked as ‘Y’ for the date the user selected from Step 5. |
| 7 | Validate/handle and notify the user of any possible errors after submit button is pressed. |
| 8 | Store user’s selected appointment date/time into a new database table called “customer”. |
| 9 | Change the “availability” column in the “appointments” table to ‘N’, that way the system knows not to display the selected date/time as available for a new user. |

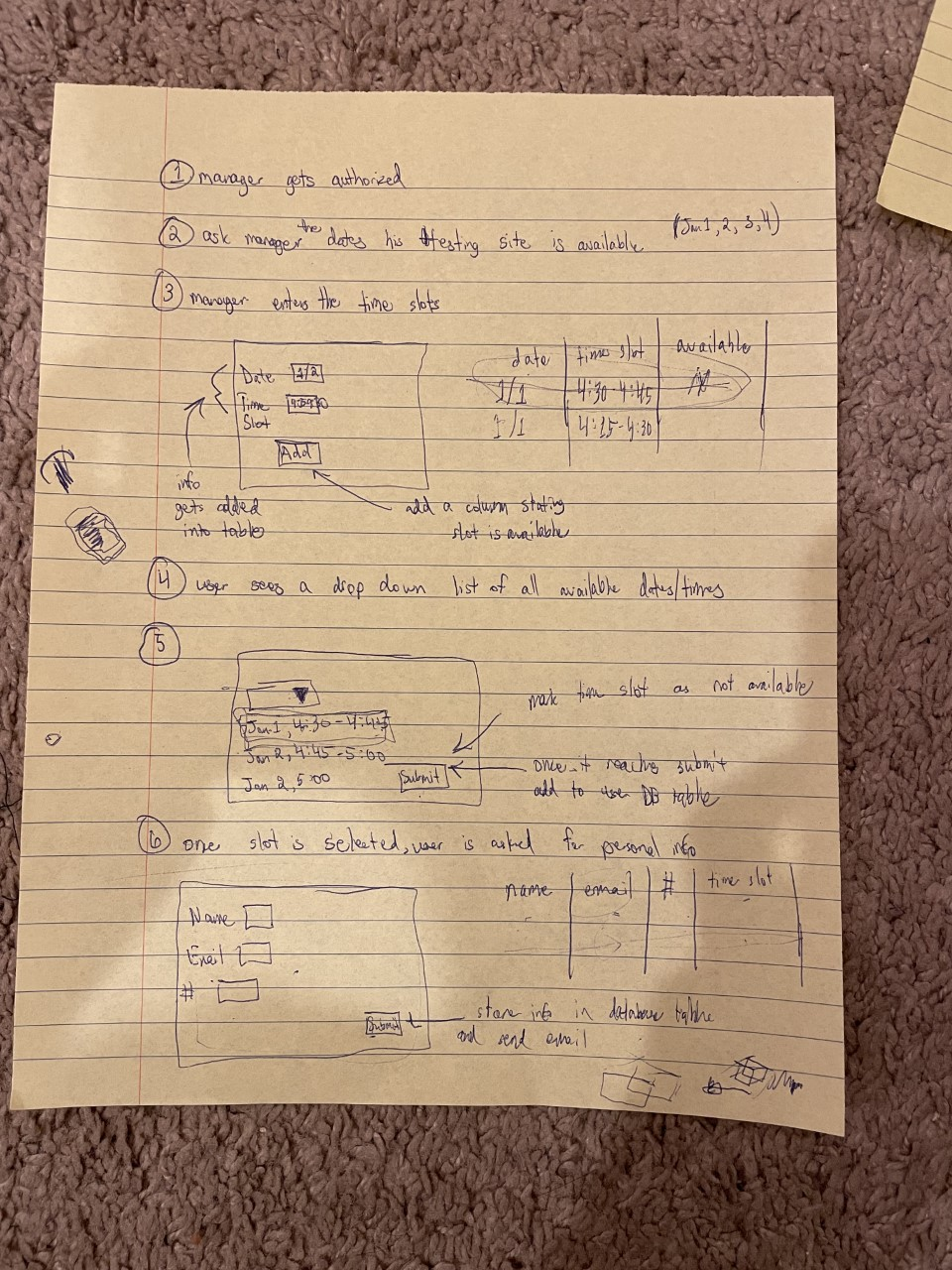
**Data Structures:**

**Version 1:**

I decided to store most of the information in a database as it would save all the necessary information even after the app gets closed/stops running. Using a database will also be helpful since I will be able to easily display and edit information through database commands in the IDE. I plan on creating objects of each database table that way I can access the database information in the memory. By creating objects of the database tables, the application would be able to run faster and more efficiently since I would not have to retrieve and get information from the database multiple times while the application is running. In order to create the database objects in the program, I will store the information by using array lists. I chose array lists instead of arrays since their size can dynamically be changed which is necessary for the application because appointment dates/times can change and their availability will also need to constantly be updated.

**UI Flows:**

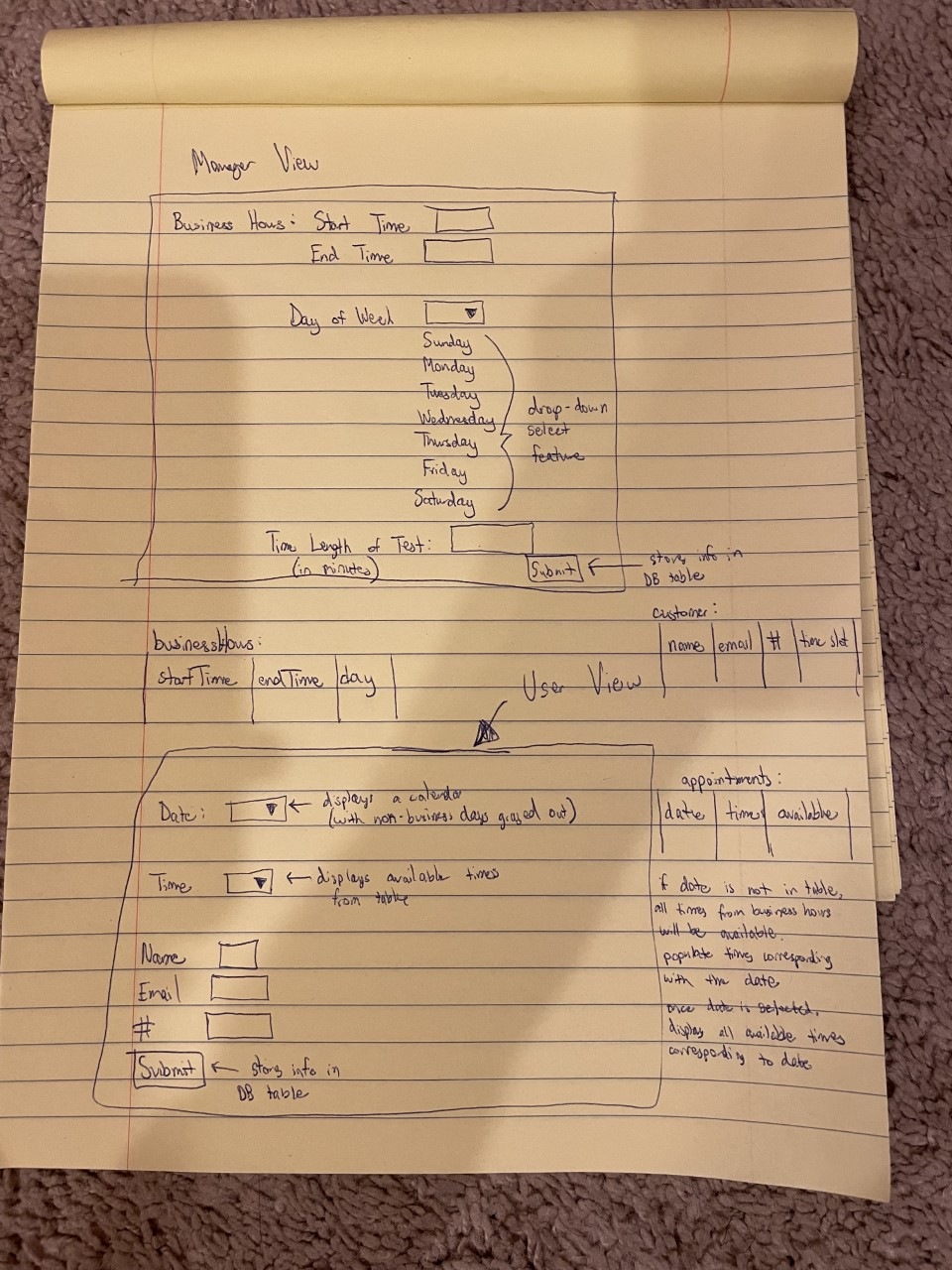
**Version 1: INITIAL ROUGH DRAFTS**



**Test Plan:**

**Version 2: INITIAL ROUGH DRAFTS**

**(WILL MAKE UI FLOW LOOK NEATER THROUGH TECHNOLOGY)**



**Test Plan:**

**Test Plan:**

**Version 1:**

| **Action to Test (Success Criteria)** | **Method of Testing** | **Expected Result** |
| --- | --- | --- |
| 1. Show a login page once directed to the app (to authorize owner) | Open the app and check if the first page which is displayed has a title of “Login” | Empty text fields for manager username and password to authorize him |
| 1. Allow owner to add, edit, and delete available appointment slots/timings from the app | Check if the various buttons/functionalities work by looking into the updated database table values | Database table values should change/update depending on the button which was pressed |
| 1. Store all the data regarding appointments and user information into the database | Check if database tables are made | Different tables with columns that accept different inputs should be created |
| 1. Display appointments that owner added to the app | Select the drop-down menu for appointments which users will choose from | Available appointment dates/times will be displayed to users |
| 1. Send a confirmation email once a user schedules an appointment | Input my personal email and check to see if I received a confirmation email | Users should receive a confirmation email regarding their appointment status |
| 1. Allow user to cancel a scheduled appointment and mark the slot available | Check if the database “availability” column changes when a user requests a change in appointment | The appointment should be marked as available and will acknowledge the user’s change |
| 1. Email the user a reminder about the scheduled appointment in advance | Input my personal email and check to see if I received a reminder about the scheduled appointment | Users should receive a reminder about their scheduled appointment 1 day in advance |
| 1. Give the manager the ability to look at key metrics | Check the manager view to see if he can view customer and appointment information | Manager should be able to view statistical information for his testing site |