Online Appointment System

SE 240 – Group 1

5/1/2020

Zac Ashburn

Caden Johnson

Yadav Bhetuwal

Sushil Khadka

Abstract

Booking an appointment online has grown in popularity over the past few years. Many different types of businesses use some type of Web-based/in person online appointment system to help make the appointments. An online appointment system allows user to make an appointment online and gives user and administrator flexibility to edit appointment. System sends notification to both administrator and user 24 hours prior to appointment and save the date and time of appointment in google calendar. This paper gives details of the development process of an online appointment system in detail.

Acknowledgement

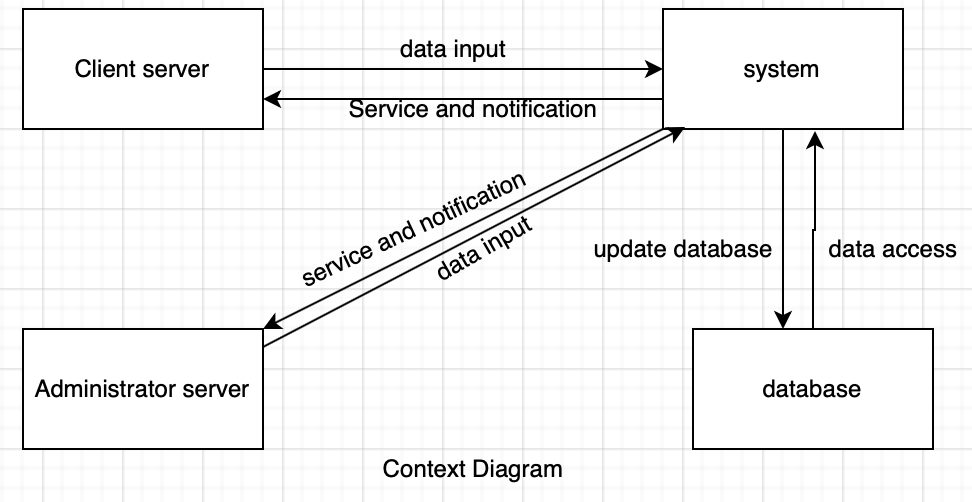
We would like to thank Dr Maninder Singh and software engineering department for providing this wonderful opportunity to work on this project and all the group member for working together and helping each other learn new things every day.

Table of Contents

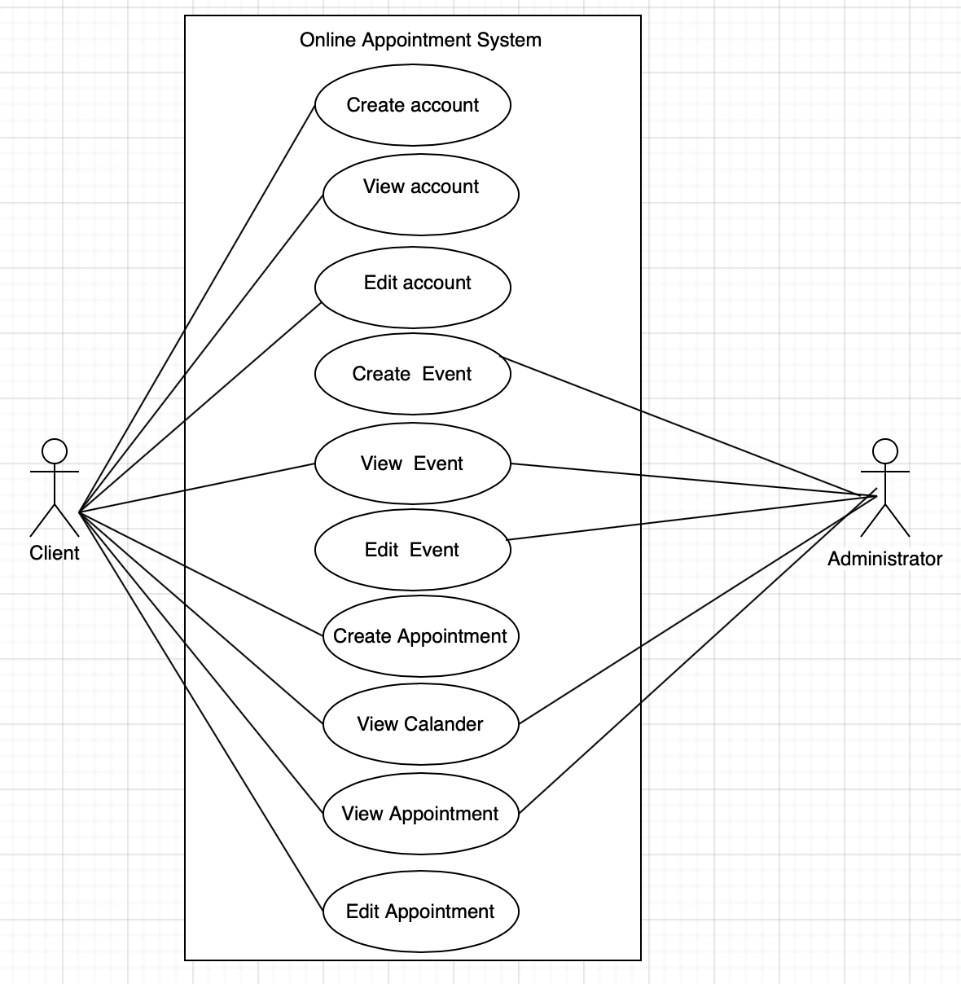
1. Title Page............................................................................................................................0
2. Abstract...............................................................................................................................1
3. Acknowledgement..............................................................................................................1
4. Table of Contents................................................................................................................2
5. List of Figures…………………………………………………………………………….3
6. Introduction………………………………………….……………………………………4
7. Background………………………………………….……………………………………4
8. Architecture Design………………………………………………………………………5
9. Component and Deployment Design……………………………………………………..5
10. User Interface Design…………………………………………………………………….6
11. Test Cases ……………………………………………………….……………………….7
12. Future improvements…………………………………………………...……………….10
13. References………………………………………………………...……………………..11

List of Figures

Context Diagram:



Use-Case Diagram:



Introduction

Problem Statement

We noticed a large amount of people who struggled to make appointments and properly manage them using reminders to keep them informed of their schedule. With an Online Appointment System, you can set your appointment for people to see and set reminders right away so if a user were to forget, a reminder has already been set.

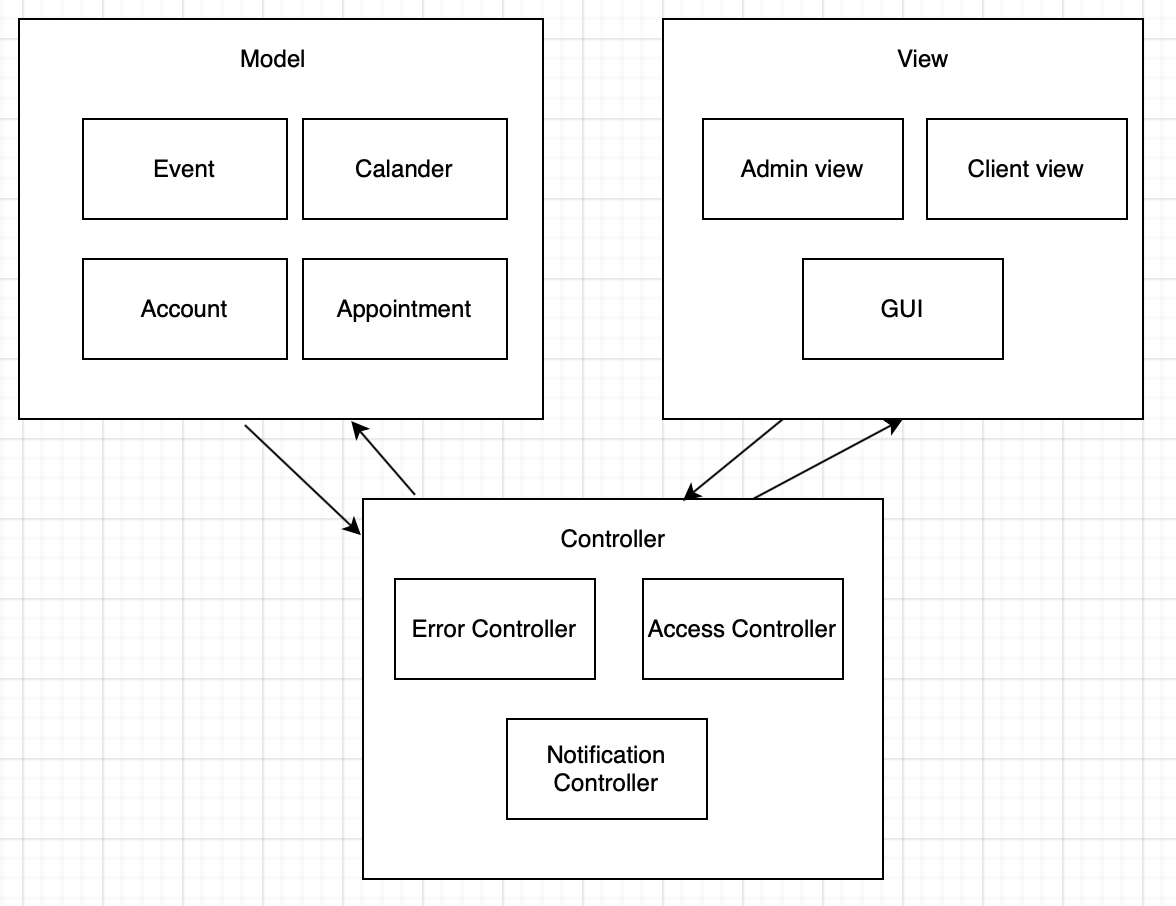
Motivation

Our motivation for this project came from a daily use of appointment systems and a familiarity of reminders/notifications. These reminders have become so common in applications, but they are rarely set up by the actual system; instead, they usually must be set up by the user. I, personally, have a bad habit of forgetting important events such as appointments, and an application/software such as this could also be useful in my daily life.

Background

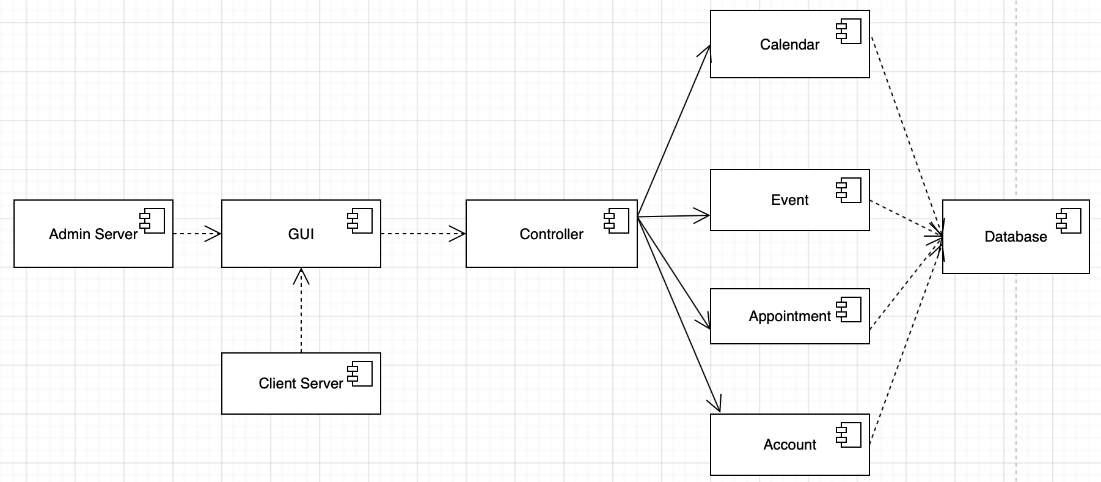
This group decided to choose this project to make in hopes that it would benefit others. The benefits being that reminders are set automatically onto a person’s phone (through the app) so they do not forget about the appointment. This will also tag into the administrator side to see his list of appointments for the day. This app provides flexibility to user to edit appointment and benefit administrator by making managed schedule.

Architecture Design

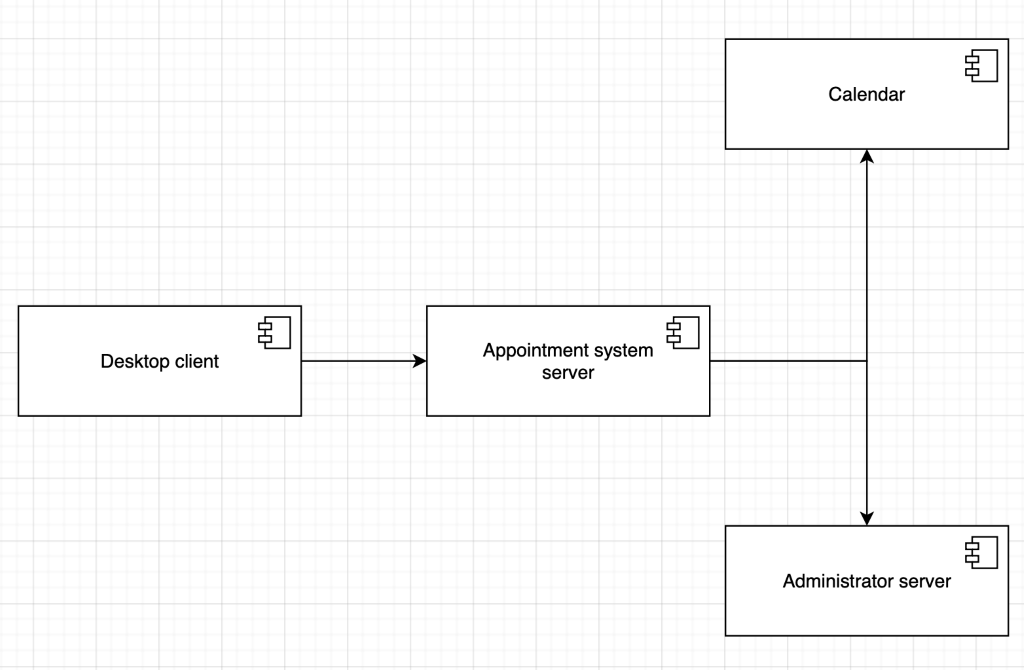


Component and Deployment Design

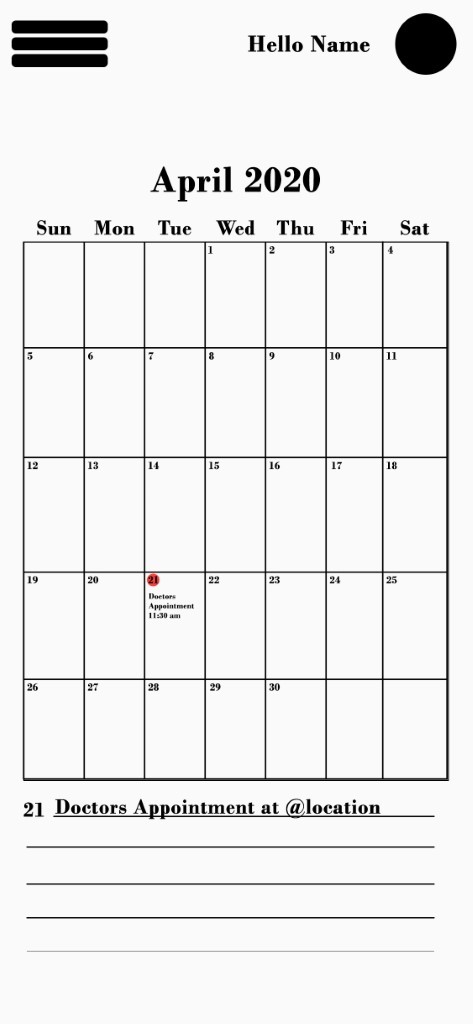
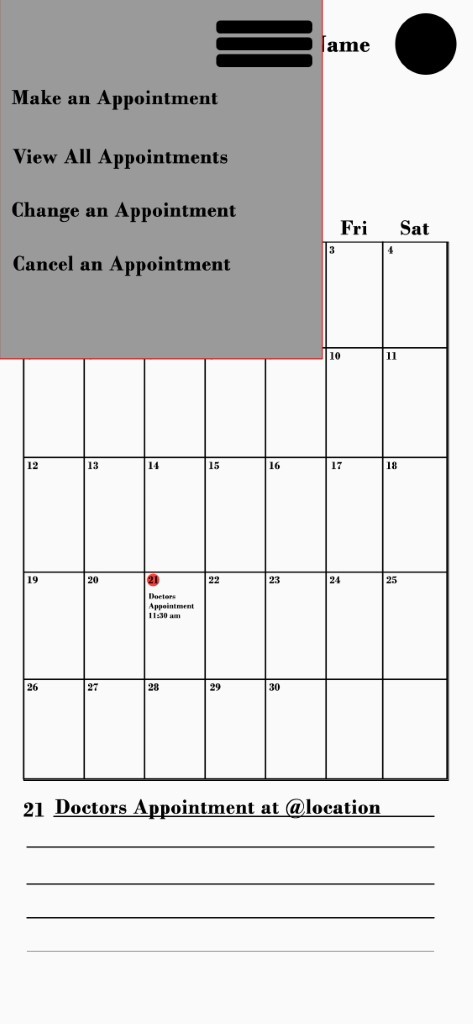
Component Design

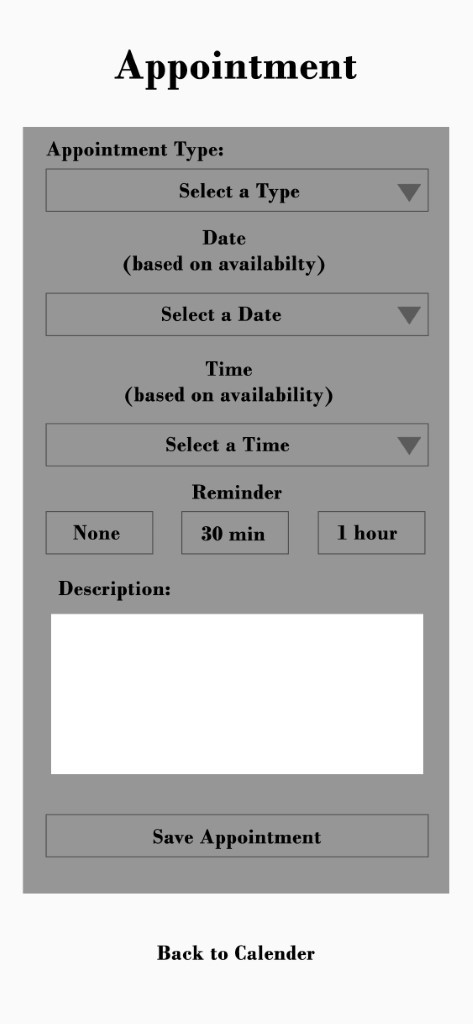
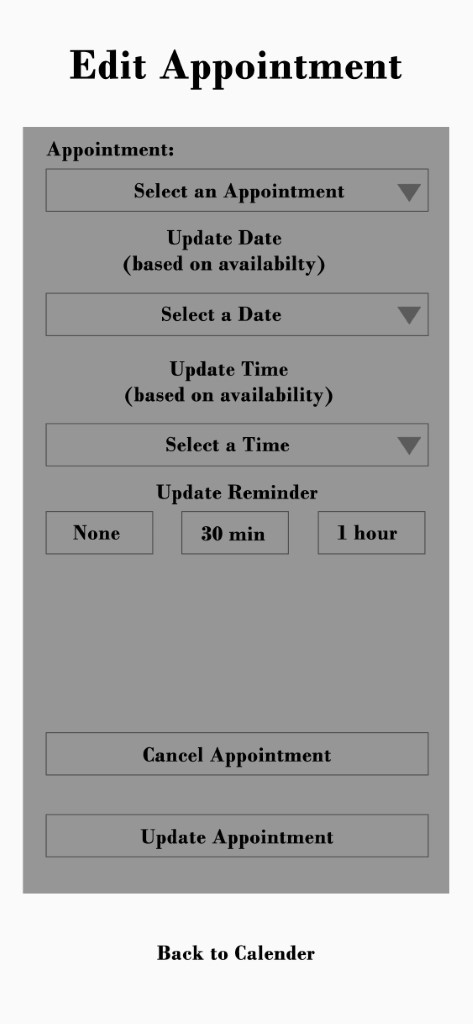


Deployment Design



User Interface Design

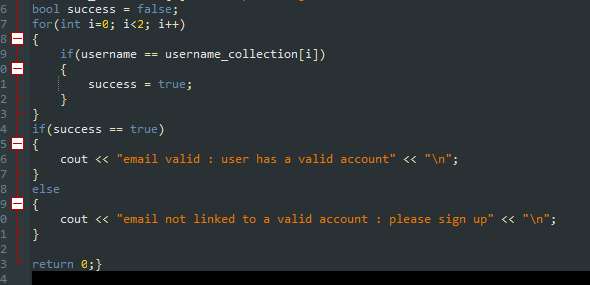
 

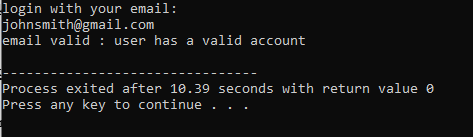
Test Cases

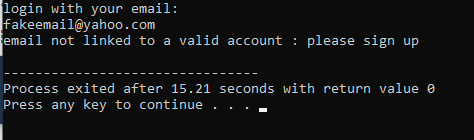
In this section we will be going over a few test cases implemented in our software. These are used to ensure certain conditions are met in order for the software to run smoothly and have limited errors. With the lack of test cases, bugs can occur, and those can lead to runtime errors, or worse: vulnerabilities for attack. As a cyber security major, I have learned the importance of reducing flaws that could become future exploits. Especially in a software that manages sensitive data like personal information and schedule times, limiting vulnerabilities is a necessity.

**Check if user has an account**

The first major test that we need to implement in our application is to test if a user trying to login has a current and viable account already registered. An example of this shown via source code and execution can be seen below:

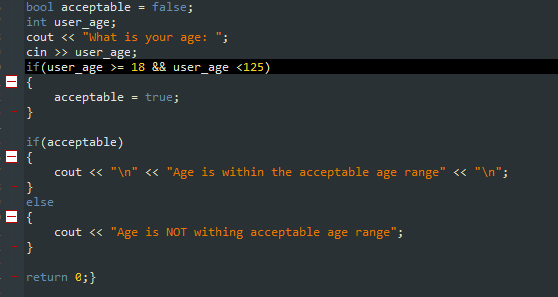


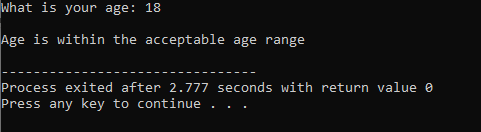




**Check if user is 18 or older**

The next thing we need to check is if the user is within the allowed age range. This should be carried out during the set-up of their new account in order to reduce redundancies further down the line. An example of this can be seen below:



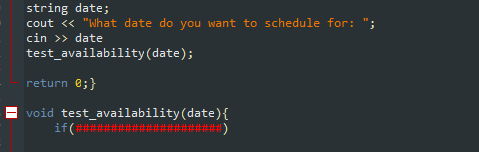






**Check if appointment time is already booked**

Lastly, we need to check whether a time that the user is trying to set up an appointment for is already taken or if it is available. This is important as to not have multiple appointments scheduled for the same time, which would just cause all sorts of confusion. An example of this can be seen below:



This test case is a bit more complicated, and I didn’t want to go too far in depth. Basically, the test\_availability function can be set up to scan the storage method used for the appointments and see if there are any scheduled during the target date/time. This was going to be a bit of coding, so I just summarized it with the simple outline.

Future Improvements

We hope to have this be implemented with your google account. This way, the calendar would be synced to your everyday calendar and you would not have to go to a separate app just to see when your next appointment is.

Additionally, future improvements may include upgrading and adding more detail to the user interface, expanding to more platforms (both mobile and desktop), and much more. The possibilities are endless with software, and an appointment system such has this has a lot of potential for growth and improvement.

References

Pressman, Roger S., and Bruce R Maxim. *Software Engineering: a Practitioner's Approach*. 8th ed., McGraw Hill., 2015.