CS 310 Project A: FindASeat

Kevin Arackaparambil, Kaitlyn Werden, Irene Park

Table of Contents

1) Team Information

Team #6

Irene Park (ijpark@usc.edu)

Kaitlyn Werden (werden@usc.edu)

Kevin Arackaparambil (arackapa@usc.edu)

2) Preface

The purpose of this document is to describe the design and requirement changes made during implementation for the client. During the implementation of the application, minor changes were made to the overall architectural and detailed design as well as general requirements. The document will also explain the rationale behind any adjustments.

3) Introduction

The system is needed in order to help students find available seating in campus buildings. Oftentimes, most of the seats in the more popular buildings on campus are fully taken, especially during heavy study weeks before midterms or finals. This means that students may have to spend unnecessary time moving between buildings or rooms trying to find available seating. The main functions of the app will be to display buildings and their number of available seats and then make, modify, and view reservations for these buildings. The app will also work with a map system/API in order to display nearby buildings on campus. The system fits into USC's overall objectives by allowing students the ability to conveniently find open study spaces in order to further their academic knowledge and skills.

4) Architectural Design Change

No changes were made to the architectural design during implementation as the three-tier architecture remained the most comprehensible and logical design as we were creating the app.

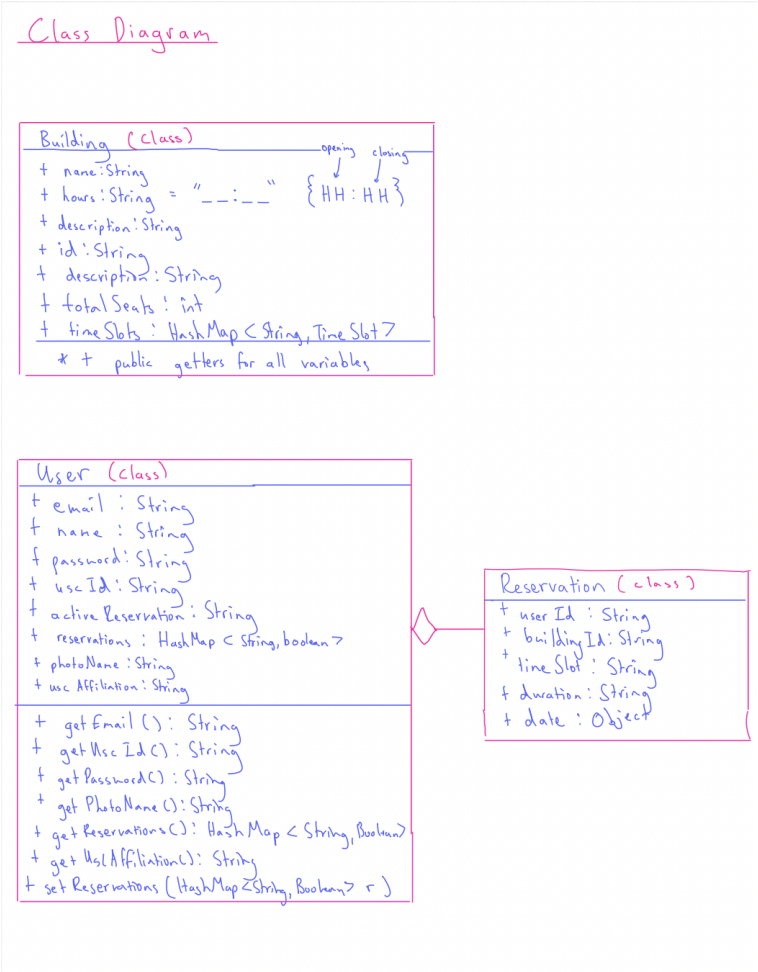
5) Detailed Design Change

Changes made during implementation and rationale behind these changes:

1) We eliminated the seat class as we determined that the actual seat a student reserves is unimportant. Instead, the student selects if they want an indoor or outdoor seat in a specific building and then has a claim over any seat in that area rather than a specific seat.
2) Our sequence diagrams were also slightly different due to the elimination of the seat class. For viewing seating availability, the user would still select a building and then view available seats in that building but rather than seeing the information and selecting a specific seat, the user just reserves a random seat in the area. Similarly, for making reservations, the user skips the step of selecting a specific open seat and can just make a general reservation. Lastly, for the sequence diagram of viewing reservation history,

the user does not need to click on “view history” as the history is automatically listed at the bottom of the user’s profile, so this step of the sequence diagram is no longer needed.

3) We continued with our three original class structures: reservation, user, and building. However, for each of these classes, we found ourselves adapting as we implemented, adding variables and methods for getting/setting variables as needed. For example, we added a timeSlots hashmap storing <String, Timeslot>. The string represents the time, such as 09:00 or 09:30. We created a subclass called timeSlot that stores the number of outdoor and indoor seats available for each building. Lastly, we created a duration variable, and a date variable in the reservation class. These variables are used to accurately determine whether a reservation is still active. The core functionality and intentions behind these classes remained the same. Below is the updated class diagram.



6) Requirements Change

We did not make changes to any requirements.