**Phase 2: Specification/Analysis Modeling and Design Modeling**

**1.** **Introduction**

**1.1 Purpose**

This specification and analysis report will illustrate the many aspects of our project we tend to implement in the form of design modeling. The planned final result of the project will be a functioning hotel reservation application that will allow users to reserve rooms, allow managers to create rooms for a given hotel, and etc. This project is being developed in the COMP 380/L class for the Fall 2024 semester. The information regarding our group and the individual contributions we have made to this phase of the project are listed below.

**Group Name: Chaotic Coders**

**Members and their contributions to this phase of the project:**

**Alex:** In this phase of the project I have worked on creating the Use-Case diagram for our application as well as implementing the Account Manager class to allow for a user to create an account, sign out of their account, and then be able to sign in using the password they created and their assigned User Number. Currently working on getting the Room class set up properly to allow Ethan to finish the Reservation Manager class. I also created the different “screens” that will later turn into our GUI windows.

**Ethan:** In this phase of the project, I worked on the backend part of reservations, and have somewhat completed one of the options under the ReservationManager class (view rooms and then make a reservation), using the flow chart made by Alex as a reference. There still needs to be some error checking done, but the basics are there. Other than that, I have made a couple edits here and there for other parts of the class and also the Reservation class. My goal is to work on completing all the options under the class in the next few days.

**Minas:** For this phase of the project, I have been experimenting with Swing to familiarize myself with the GUI framework. Nura and I collaborated to divide tasks. I’ve already developed a simple working example and am now working on implementing it within the context of our current project. Current goal is to get a demo example within next week and see what we build on it.

**Nura:** In this phase, I and Minas are in discussion on how we will build the front end of the project. I am still learning how to create a GUI. Therefore, I am watching SWING tutorials and Java FX. My goal is to master these softwares and also learn how to use WindowBuilder.

**1.2 Definitions, Acronyms, or Abbreviations**

**User**: A user refers to any person who uses our application.

**Manager**: A user who is given additional privileges to operate the hotel and manage reservations.

**1.3 References**

We have not used any external sources for referencing during this development stage of the project so far. Everything has been original work.

As for sources that we are currently using in our project and have made ourselves, we are primarily going off of the flow chart graph made by Alex which is displayed below in the ‘overview’ section. It includes the classes we will implement and their functionalities, as well as the procedures of the application.

These are other documents that are stored in our repository for documentation purposes, among the other project topic forms that we have already submitted:

**Main States:** <https://github.com/Alexander-Boutselis/Chaotic_Coders/blob/main/Documentation/Main%20States.png>

*A simple image of the main states for the program.*

*October 11, 2024*

**Sprint 1 CRC:**

<https://github.com/Alexander-Boutselis/Chaotic_Coders/blob/main/Documentation/Sprint%201%20CRC.png>

*Our CRC for Sprint 1.*

*October 11, 2024*

**Hotel App Flow Chart:**

<https://github.com/Alexander-Boutselis/Chaotic_Coders/blob/main/Documentation/Hotel%20App%20Flow%20Chart.pdf>

*A flow chart created by Alex that we are referencing.*

*This was made using Lucidchart. The direct link can be found below in section 1.4.*

*October 9, 2024*

We may be using external databases (possibly AWS) to implement where we store our data, since naturally, a hotel reservation system must hold all the reservations, rooms, and users. However, this is not completely finalized yet.

**1.4  Overview**

**2.** **Use-Case Modeling**

**A diagram of a flowchart

Description automatically generated**

<https://lucid.app/lucidchart/bdd7e834-0026-44b3-a36d-3c389f4d5df6/edit?view_items=Ov9fxYa2L-As&invitationId=inv_863ba3e0-3c0d-40cc-9985-8dd461fb8803>

**3. Class Modeling**

**3.1** CRC Model for our ***first sprint*** from our user stories.

A black board with white writing

Description automatically generated

3.2 Construct UML Class Diagrams (include any sub diagrams) from CRC Model for your first sprint. Provide all instance data members and methods for your classes.

A diagram of a company

Description automatically generated <https://lucid.app/lucidchart/e6634381-65d1-48e0-afe2-a4c890a10c07/edit?beaconFlowId=0689EFD814D94385&invitationId=inv_9c5d8b70-e5e2-4665-a732-b098144e28d1&page=HWEp-vi-RSFO#>

**4. Database schema**

**(Database schema is still a work in progress as we get the functionality of the application working)**

Entities:

1. **Hotel**:
   * hotel\_id (Primary Key)
   * name
   * total\_rooms
   * current\_user\_id (Foreign Key referencing User)
2. **Room**:
   * room\_id (Primary Key)
   * hotel\_id (Foreign Key from Hotel)
   * room\_number
   * room\_status (Tracks whether a room is available, reserved, etc.)
3. **User**:
   * user\_number (Primary Key)
   * first\_name
   * last\_name
   * password
   * birthday (This field is inferred from the Database class, and is of type Calendar)
   * current\_reservations (Foreign Key from Reservation)
4. **Reservation**:
   * reservation\_id (Primary Key)
   * user\_id (Foreign Key from User)
   * room\_id (Foreign Key from Room)
   * start\_date
   * end\_date
   * reservation\_status (Stores whether the reservation is active or completed)
   * total\_price

Relationships:

* **Hotel** ← hotel\_id → **Room**
* **Room** ← room\_id → **Reservation**
* **User** ← user\_id → **Reservation**