**Software Design Document**

April 5, 2019

Team 2 (Dimitrios Chavouzis, Claire Collver, Clarissa Fung, Drew Hager, Blake Skelton)

**Introduction:**

This report describes Team Software Design for our project: a Space Reservation System

1. Introduction (generally similar to the introduction in your architecture document)

1.1. Purpose

The purpose of this project is to generate a system that allows students at Davidson College to reserve locations for the purpose of meetings and/or private study.

1.2. Scope

What is covered by this document. E.g., does it cover the entire project, or just the changes from some other product?

The scope of this document covers the entire space reservation system project.

1.3. Definitions, Acronyms, and Abbreviations

HTTP – Hypertext Transfer Protocol

JDBC – Java Database Connectivity

SQL – Structured Query Language

C&C – Components and connectors

1.4. References

List any documents referenced by this document. Include standards, technical documents, etc.

No references are made in this document.

1.5. Overview

Give a very high level description of the project. A couple of paragraphs will do. Serves to give a context for the next section Overall Description of the specification

Our space reservation system will have two main functionalities: the ability to check the occupancy status of a room given the name of its building; and the ability to reserve a specific room for a certain amount of time. We will use a website for our platform and MySQL for our database. The database will need to be populated with room numbers, capacities, building names, attributes, utilities, etc.

2. Software Design Description

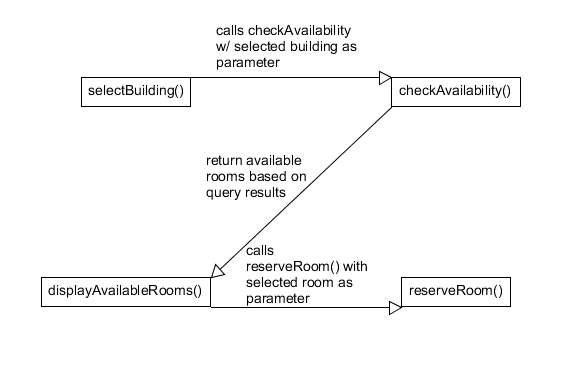
2.1. Client

2.1.1. Design Overview

We are performing a functional design. The overall function of this is for users to reserve a room or check availability of a room. This component facilitates the transformation of user needs into database queries and communicates that database query to the server.

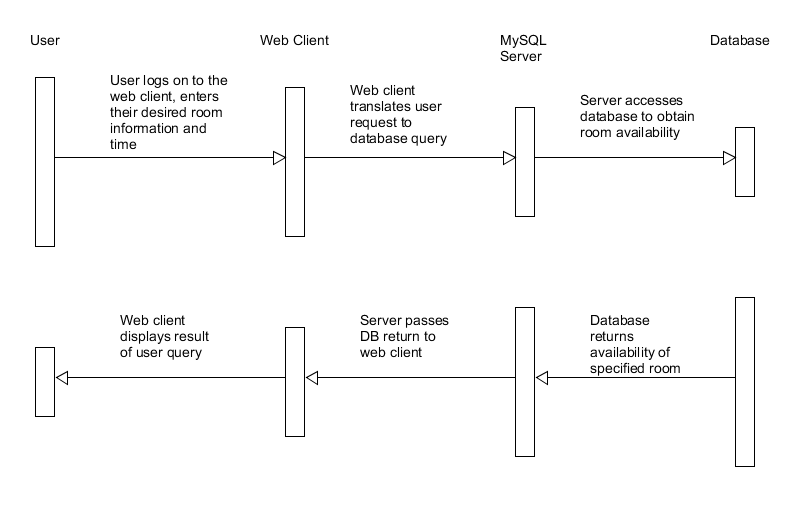
2.1.2. Javascript/HTML/PHP

2.1.3. Class diagram or data structure diagram



2.1.4. Sequence diagram for key use cases

Use case: User would like to reserve a specific room at a specific time:

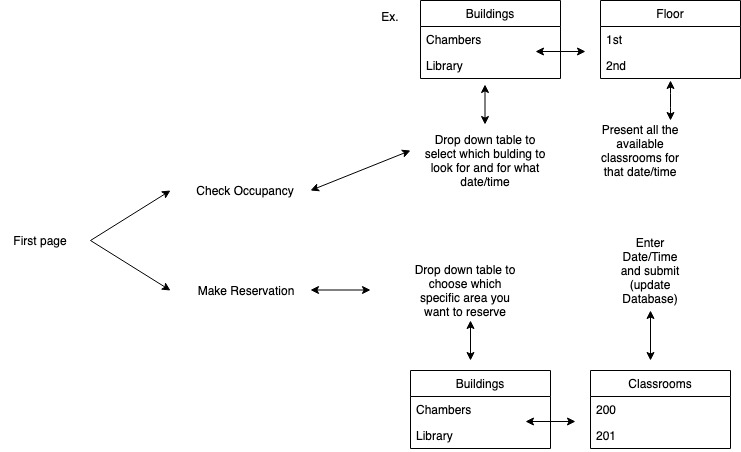


2.1.5. Detailed Design

2.1.5.1. Logic/Algorithm Design

1. Prompt user to decide between reserving a space and checking availability.
2. If checking availability
   1. Prompt user to select desired building.
   2. Request data from server about number of available rooms per building.
   3. Display number of available rooms per building.
   4. Allow user to click on building names to see specifically which rooms are available.
3. If reserving room
   1. Prompt user to click dropdown boxes for date, time, building, room.
   2. Request data from server regarding the selected inputs.
   3. Check availability of selected room at specific time. If available change occupancy status to reserved, else display room not available message.

2.1.5.2. State Diagram (if applicable)



2.2. Server

The overall function of the server is to translate queries received via connection with the web client into responses from the database itself. We will be using a MySQL server hosted by Davidson College. We will trust the Database Management System to handle queries from the database connection and optimize the return of results. This component of the project does not require development on our part.

2.3 Database

2.3.1. Design Overview

The function of this component is to store and return information about reservable rooms on campus as well as their attributes. MySQL is a relational database

2.3.2. MySQL

2.3.3. Class diagram or data structure diagram

We will create a dataImporter class that will parse information about the buildings and populate the database.

2.3.4. Sequence diagram for key use cases

DBMS will handle sequence of operations/flow of control when fetching queries.

2.3.5. Detailed Design

2.3.5.1. Logic/Algorithm Design

The DBMS uses mergeSort to sort entries based on a secondary index, such as when rooms are sorted by location in the building.

3. Inter-component or inter-subsystem communications

3.1. Client-Server

3.1.1. Interface description. Include maximum data rate, message format (if applicable), response format (if applicable), error handling

The webapp will make requests to the server when it needs to query the database. Maximum data rate is low. Queries that are conflicting with other reservations will return an error to the webapp, which will in turn display a message to the user that this time is not allowed.

3.2. Server-Database

3.2.1. Server communicates with database to query the database. Low data rate because queries are infrequent. If the reservation that the user is trying to make conflicts with another reservation the database will respond with an error message which will be displayed by the web app.

4. Metrics

4.1. Size information

We don’t expect our code to exceed a thousand lines of code (1 KLOC). We have three modules.

4.2. Complexity

4.2.1. Network Metrics

Graph Impurity = n - e - 1 = 3 - 2 - 1 = 0

We have three modules and two connectors between them. Using the graph impurity formula we expect our graph impurity to equal zero.

4.2.2. Information Flow Metrics

Module design complexity = size \* (inflow \* outflow)^2

= 1 KLOC \* (1\*1)^2

= 1 KLOC