**Software Requirements Specification**

February 28, 2019

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

**Introduction:**

This report describes Team Software Requirements for its project to develop a space reservation system for Davidson College.

The format will follow the format shown in the textbook, Chapter 3, Section 3.3

1. Introduction

1.1. Purpose

The purpose of this project is to create a space reservation system for Davidson college where users can reserve rooms on campus for studying, club meeting, etc...

1.2. Scope

This document covers requirements necessary for completing all four iterations. However, since we are using an iterative model, we expect possible future additions or updates to these requirements.

1.3. Definitions, Acronyms, and Abbreviations

* DBMS is the Database Management System
* JDBC stands for Java Database Connectivity

1.4. References

* Documentation for MySQL: [*https://dev.mysql.com/doc/*](https://dev.mysql.com/doc/)-
* JavaScript reference documentation: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference>

1.5. Overview

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.

Our project will have four iterations and incorporate user stories in order to revise our requirements. We will use focus group test sessions to find bugs in our project and ensure our application is user-friendly and pair programming to eliminate bugs on the spot. The use of sub teams will help us to divide and conquer where our skill are most applicable.

2. Overall Description

2.1. Product Perspective

The hardware we will be using is our laptops as we do not have a budget to purchase any other hardware. Since we are creating a website, the operating system will be whatever system users choose to access the site. Most of our computers run on Mac OS X (UNIX operating systems). We will likely use MySQL to create our database, as our team is already familiar with MySQL.

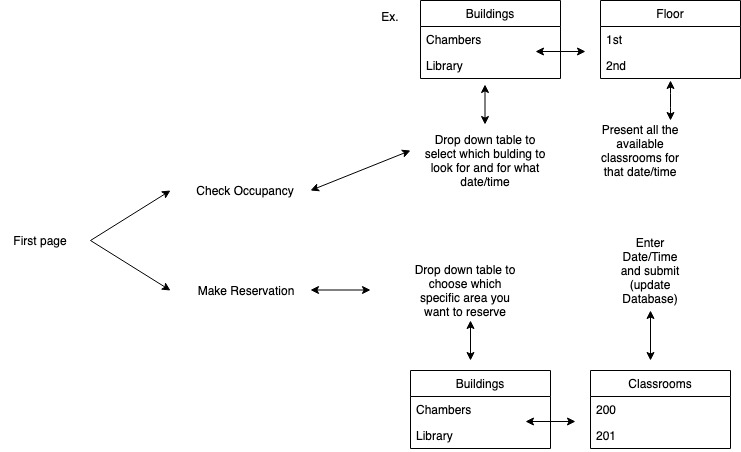
**Use case walkthrough** - a student logs on to the web client, enters their desired room information and time. The website accesses the remotely stored DBMS, retrieves the appropriate information, and returns the room availability to the user. Based on that information, if the user decides to reserve a room, the web client generates a query to enter the reservation into the system.

**Secondary use case walkthrough** - a student logs on to the app to check if their favorite study room is available. They select the correct building and floor. The app accesses the database and generates a graphic representing the floor with available rooms in green. The student determines that their favorite room is currently in-use and decides to stay home.

These use cases fit into the larger context of the Davidson community in that our users are Davidson students participating in activities integral to their lives as students.

2.2. Product Functions

There are two major functions, occupancy check and room reservation.



2.3. User Characteristics

The users are young, motivated students that require rooms for a multitude of reasons at a specific time. Students may need rooms to study in, hold club meetings, or host speaker events. Most students at Davidson walk as their main form of transport. They also should be interested in using the application.

2.4. General Constraints

The product database will have to live on Davidson’s domain so that it will be able to be accessed independently of our computers.

2.5. Assumptions and Dependencies

1. We will be able to obtain a stable place to host the database.
2. We will be able to populate the database.
3. We will learn the skills necessary to complete the project.

3. Detailed Requirements

3.1. External Interface Requirements

3.1.1. User Interfaces

* Use of dropdown boxes for users to choose the building options
* To reserve:
  + A screen where users can select the date and time and duration they wish to reserve the space for
  + Finally a screen where available rooms are listed and the user can select which one they wish to reserve/enter their Davidson ID to confirm
* To check occupancy:
  + A screen that allows users to select the floor they want to check
  + A screen that shows a list of available rooms at the current time as well as an option to search at future times

3.1.2. Hardware Interfaces

* None

3.1.3. Software Interfaces

* MySQL
* Visual Studio

3.1.4. Communication Interfaces

* MySQL and website (JDBC)

3.2. Functional Requirements

*You will list each major item in your product and the requirements for each. For each actual requirement, use the word “shall” if the implementation must do it, “may” or “should” if it’s not mandatory, but just desirable. Most requirements should use “shall.” Remember, every requirement must be verifiable. Check that you have met the 6 characteristics in section 3.3.1 of good requirements – correct, complete, unambiguous, verifiable, consistent, ranked for importance and/or stability.*

*If your project is using use cases, you can replace the corresponding functional requirements in this section with use case descriptions.*

3.2.1. Database shall be populated with data for the library, Sloan, and

Wall academic buildings.

3.2.1.1. Shall retrieve appropriate room information as requested

by the web client or app

3.2.1.2. Shall represent room number, building name, capacity,

utility, availability, and occupancy data for all desired

buildings

3.2.2. Database shall check occupancy for the selected room at the selected time and return a boolean. (1 if occupied, 0 if not occupied)

3.2.2.1. Shall present dropdown boxes with options for building,

floor, time, and duration.

3.2.2.2. Shall read information from the database to determine

available rooms at a specified time.

3.2.2.3. Shall present the user with a list or graphic of the

available rooms on the specified floor for the specified

time.

3.2.3. Database shall make a reservation for the selected room and for

the selected time by updating the database with boolean values.

3.2.3.1. Website shall present dropdown boxes with options for

building, room number, time, and duration.

3.2.3.2 Database shall be updated with reservation

information if room is available.

3.2.4. Performance Requirements

Response time should be immediate (as perceived by users).

3.4. Design Constraints

Since we only expect to populate our database with three buildings, we don’t expect memory or file size to be constraints.

3.5. Attributes

When a reservation is being made, the system shall prompt the user for their student ID number, which shall be stored in the database with the reservation.

3.6. Other Requirements

Members of the team may receive food and coffee every Sunday during team meetings.