Team Final Report

May 3, 2018

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

Introduction:

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

1. Project Management

Consider your project management report, questions 1, 2, and 4. Compare what you said then with your actual results. How accurate were your sizing estimates? How accurate was your high-level schedule? Did additional problems develop that weren’t on your list of risks? How effective were your risk mitigation strategies?

We did not require as many lines of code as we originally anticipated. While we did spend about 7.9 person-months working on the project, a great deal of our time was spent problem-solving and determining the functional requirements of our project. We did not formally delimit our iterations, but we did create several versions of the website and the database as we continued to focus in on our goals. While we were unable to use the EMS database, we resolved this issue by gathering data ourselves. Other problems we encountered mostly included issues with MySQL functioning normally as well as integration with the website. Our risk mitigation strategies utilizing the expertise of professors on campus helped us to resolve these issues.

Previous answers:

* Expected level of effort in person months: 7.9 person months (~1400 loc)
* Overall high level schedule: Four iterations, spanning 2-3 weeks each. A total of 10 weeks max because of time constraint. (See number 6 for detailed schedule)
* Risk management plan:

1. Underestimated effort required due to lack of experience
   1. Probability: high (we do not have the skills to ensure that our schedule is viable)
   2. Loss: medium (we may not implement every iteration we currently have planned)
   3. Mitigation Plan: If necessary, we can eliminate the last iteration and spend our time completing earlier iterations.
2. Users do not use the system to reserve rooms or update occupancy status
   1. Probability: high (more time and institutional support would be required for widespread adoption)
   2. Loss: medium (without users filling in occupancy status of rooms, our site will not be able to provide accurate data)
3. We cannot obtain information to populate the database
   1. Probability: low (the data we require is fairly accessible)
   2. Loss: high (without a working database, our project cannot function)
   3. Mitigation Plan: We will make connections with the Davidson staff in charge of EMS and ensure that we will have access to the required data. In a worst-case scenario, we could attempt to use the database associated with EMS directly.
4. Davidson prefers that study rooms remain first-come first-serve
   1. Probability: high (the current first-come first-serve system is working fine)
   2. Loss: low (our project can function regardless of institutional support)
   3. Mitigation Plan: We could potentially market our project to other institutions
5. Speed - the database is prohibitively slow for use in the web client/app
   1. Probability: low (DBMS is optimized for speed)
   2. Loss: medium (users will be less likely to use a frustratingly slow interface)

2. Process Model

Consider your process model report, questions 1, 2, and 5. Compare what you said then with what you actually did with respect to your development process. What worked, what didn’t? If you didn’t actually do what you planned, why not?

While we originally planned to use the spiral process model, we instead used the iterative model with some agile processes as well. Iterations included 1) creating a barebones web app and database that are synced, 2) ensuring that “check occupancy” can toggle between buildings, and that 3) “make a reservation” can successfully write to the database.

The iterations were good goals, but the timeline we created for them was not accurate. We should have created an earlier iteration that did not involve linking the website and the database, which could have been achieved early. Linking the database and website was not achieved until later in our project because of the work required in setting up these structures first. Additionally, while we planned to not work on the visual aspects of the website until after it was functional, we worked on these throughout the project to ensure ease of use.

Previous answers:

* We plan to use the spiral process model as our primary process model.
* We plan to implement parts of the agile model throughout the stages of the spiral process. In particular we hope to incorporate face-to-face interactions and requirements flexibility. Additionally, we are considering using user stories to help create our requirements.
* We didn’t consider any other process models because they all had an aspect that we didn’t like for this project. Timeboxing is too complicated for our 5 person team. The waterfall model requires that all requirements are established at the beginning of the project process, but we won’t know all of our requirements until later in the process. We are using parts of the agile process later in our project. The rational unified process is considered more of a meta-process and we want a more detailed process that can take us through the entire project from start to finish.

3. Software Requirements

Consider your SRS, paragraphs 2.1, 2.2, and 3.2. Compare the requirements you wrote with what you built. List the requirements you didn’t include at all, as well as requirements you included, but didn’t complete, if any. Why were some requirements not met? Were there new requirements you included, but weren’t in your SRS? If so, how did they arise?

Our requirements surrounding the hardware were accurate, as we did not anticipate needing hardware other than our laptops to host the database and web client. Our projections regarding our use of MySQL were also correct, as were our functional requirements for the database and the web client. One requirement that we did not meet was the graphical component. We originally wanted to display a floor-plan to users selecting rooms, but that would require image-processing on the floor plans. Another requirement we did not include was the data from the Wall building, as we felt that our time was better spent manipulating the data that we had. We met all other requirements.

Previous answers:

* 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.

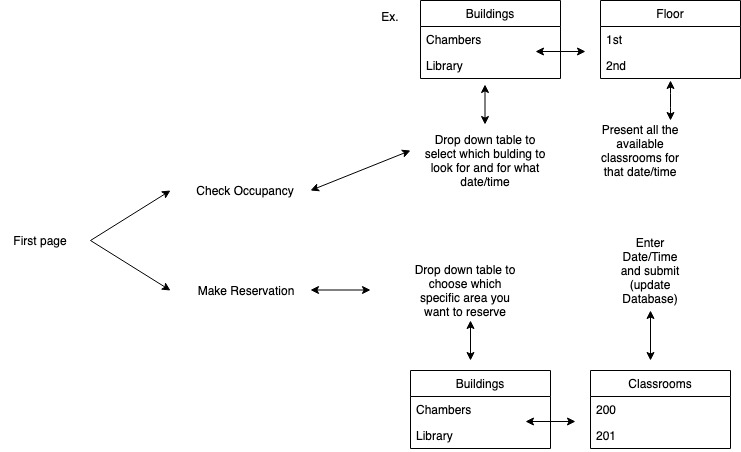
**Use case walkthrough** - a student logs on to the web client, enters their desired room information and time. The website accesses the remotely stored DBMS, retrieves the appropriate information, and returns the room availability to the user. Based on that information, if the user decides to reserve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

* 3.2 Functional Requirements

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).

4. Software Architecture

Consider your software architecture specification, paragraph 2.1. Did you implement the same set of components you described there? List any additional components you needed, as well as any components you included in the architecture specification but that you didn’t actually build.

The components described in our software architecture specification are accurate. Our final project consisted of a web client which communicates via http to a server, which uses JDBC to communicate with our database.

Previous answer:

* 1. Views

C&C Diagram:

5. Software Design

Consider your software design document, sections 2.1 - 2.n, where n is the number of components you described there. Without going into a lot of detail, how close is your actual design to the design you wrote there. On a scale of 1-10, where 10 means the design didn’t change at all, rate how closely you followed the design. Explain any discrepancies – I’m not looking for a lot of detail, but what general issues resulted in your not following the initial design?

Our web client does use Javascript, HTML, and PHP to translate user needs into database queries, which are communicated to the server. The structure of our class diagram does not exactly reflect the structure of our final project, but the flow of data is largely the same. We do plan to add an intermediary step between the retrieval of available rooms and the creation of a new reservation that will allow users to confirm their room and time selection, but this is not a significant change to our design.

Previous answers:

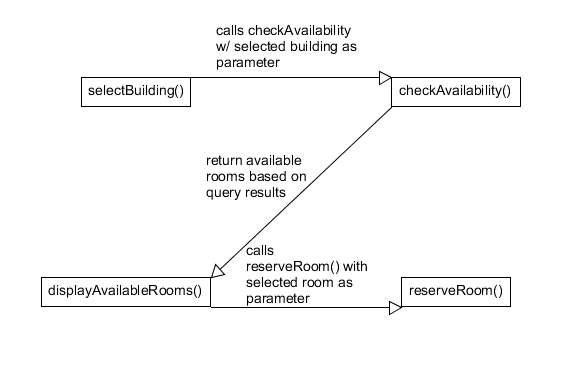
2.1. Client

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.

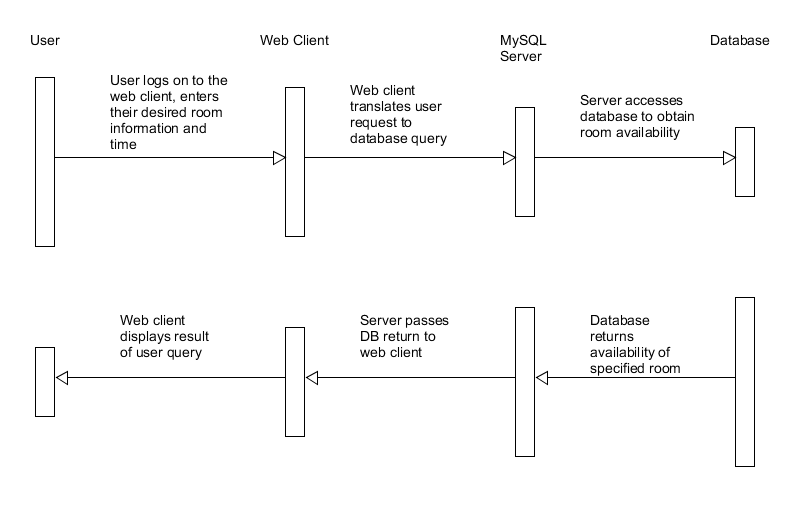
2.1.2. Javascript/HTML/PHP

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:

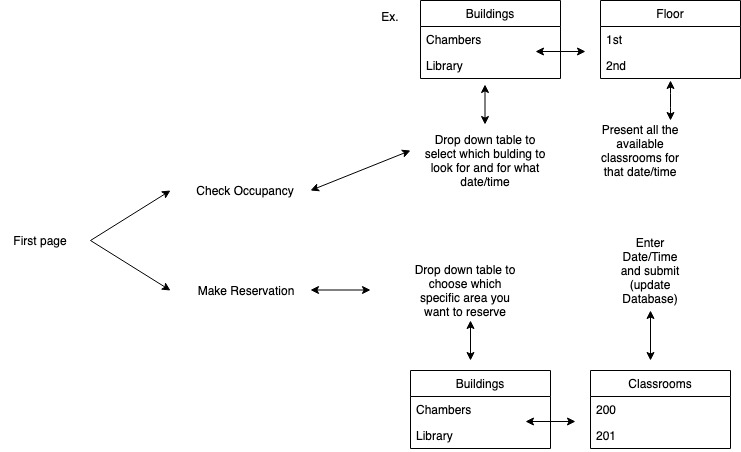


2.1.5. Detailed Design

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.

6. Summary

How would you describe this team project in terms of success? Was it fun for the team? Do you think you will do anything more on it after the semester is over? Was it easier or harder than you thought it would be? Would you recommend a future team do a similar project? Please state why or why not for each of these questions.

We consider our project to be a success. While we had to modify our schedule and milestones, by the end of the semester we managed to create a functional space reservation website that connects to a populated database. Users can select a room from the provided study rooms and reserve it for a certain time or check its occupancy. If we had more time to expand our database, or gain access to an existing database of campus rooms available for student activities...

Overall our team thought the project was fun. We got to know each other through our frequent team meetings outside of class. Furthermore, each member of our team learnt important software development skills, gained collaboration experience and practiced presenting demonstrating our work. We expect these skills to be very useful in our future careers.

We don’t think continuing the space reservation system after the semester ends is viable since we all have busy schedules. Further development of this project would be too costly for us, so we don’t expect to work on this project after the semester. However, we do see potential in repurposing our code towards other future projects.

The project was harder than we anticipated. It was much more challenging to adhere to a schedule than we had expected. We ended up needing a lot more intermediate steps and occasionally ran into bugs that halted productivity on the entire project. However, we would still recommend this project to future teams. The project was challenging yet feasible.