**Final Report**

**Instructor-Course Assignment Application**

**Project Manager: Kiana McDaniel**

**David Johnson, Phillip Tran**

**Texas State University**

**Ingram School of Engineering**

**Texas State University**

**601 University Dr**

**San Marcos, Texas, 78666**

**05/01/2019**



# Overview

## Executive Summary

Our senior design project was sponsored by Dr. Rich Compeau and Texas State University, our team consisted of three computer engineering majors. Our project’s purpose was to provide program coordinators at Texas State University’s Ingram School of Engineering with a tool that would allow them to more easily perform instructor-course assignments. The solution decided on was a server-based program that could be accessed through a webpage to allow for multiple users, remote access, and the ability to store and retrieve data as needed. The project was overall successful, the final product does everything it was intended to, albeit not perfectly and it can be improved upon. The RESTful API design worked very well to provide a system with low latency and low memory usage, SQLite worked well to provide a simple database solution although it may need to be replaced as the data size and number of tables increases. Using three different systems, and a total of five different languages made the project more complex than it needed to be and is probably the biggest detriment to the system.

*NEW PARAGRAPH:Describe the purpose & value of your project. Why did you do it? Who benefitted? How is it of value to your Sponsor/TXST/society/you, etc?*

## Abstract

The nature of instructor-course assignment is tedious and challenging due to the amount of information needed to make each assignment. With our Senior Design Project, we aimed to develop a web-based application (coursebrew) for Program Coordinators that simplified the process of instructor-course assignment. In addition to providing an intuitive user interface, coursebrew makes instructor-course recommendations and warns the user when different assignment criteria are not met or over met. The software architecture used to build coursebrew is modular, which allows future developers to easily manage and scale the system.

From page 2 forward, this document shall have:

* 1” margins all around
* Right-justified or not is your choice
* Times New Roman 12 point (EXCEPT section headers) for text
* 1.15 spacing
* 0 points before, 0 points after
* Use block separation for paragraphs - NOT indentation

Delete this text box once you have understood and obeyed its commands.

Contents

[1 Overview 2](#_Toc509991144)

[1.1 Executive Summary 2](#_Toc509991145)

[1.2 Abstract 2](#_Toc509991146)

[2 List of Figures 3](#_Toc509991147)

[3 List of Tables 4](#_Toc509991148)

[4 Problem Description 4](#_Toc509991149)

[5 Progress Towards A Solution 5](#_Toc509991150)

[5.1 Design Decisions 6](#_Toc509991151)

[5.2 Design Approach 6](#_Toc509991152)

[5.3 Project Approach 6](#_Toc509991153)

[5.4 Engineering Standards 6](#_Toc509991154)

[5.5 Progress Towards Goals 7](#_Toc509991155)

[5.6 Verification 7](#_Toc509991156)

[5.7 Characterization Results 7](#_Toc509991157)

[5.8 Deficiencies 8](#_Toc509991158)

[5.9 Iterations and Redefinitions 8](#_Toc509991159)

[6 Constraints 9](#_Toc509991160)

[6.1 Budgetary 9](#_Toc509991161)

[6.2 Design Feasibility 9](#_Toc509991162)

[6.3 Manufacturability 9](#_Toc509991163)

[6.4 Maintainability 9](#_Toc509991164)

[6.5 Environmental 9](#_Toc509991165)

[6.6 Health and Safety 10](#_Toc509991166)

[6.7 Social 10](#_Toc509991167)

[7 Budgets 10](#_Toc509991168)

[8 Work Schedule 10](#_Toc509991169)

[9 Personnel Interactions 11](#_Toc509991170)

[9.1 Teamwork 11](#_Toc509991171)

[9.2 Mentorship 12](#_Toc509991172)

[10 Ethics 13](#_Toc509991173)

[11 Summary & Conclusions 13](#_Toc509991174)

[12 Discussion 13](#_Toc509991175)

[12.1 Academic Preparation 13](#_Toc509991176)

[12.2 Lessons Learned 13](#_Toc509991177)

[12.3 Soft Skills 14](#_Toc509991178)

[12.4 Schedule Deviations 14](#_Toc509991179)

[12.5 Staffing 14](#_Toc509991180)

[12.6 Final Observations 14](#_Toc509991181)

[13 Acknowledgments 14](#_Toc509991182)

[14 References 14](#_Toc509991183)

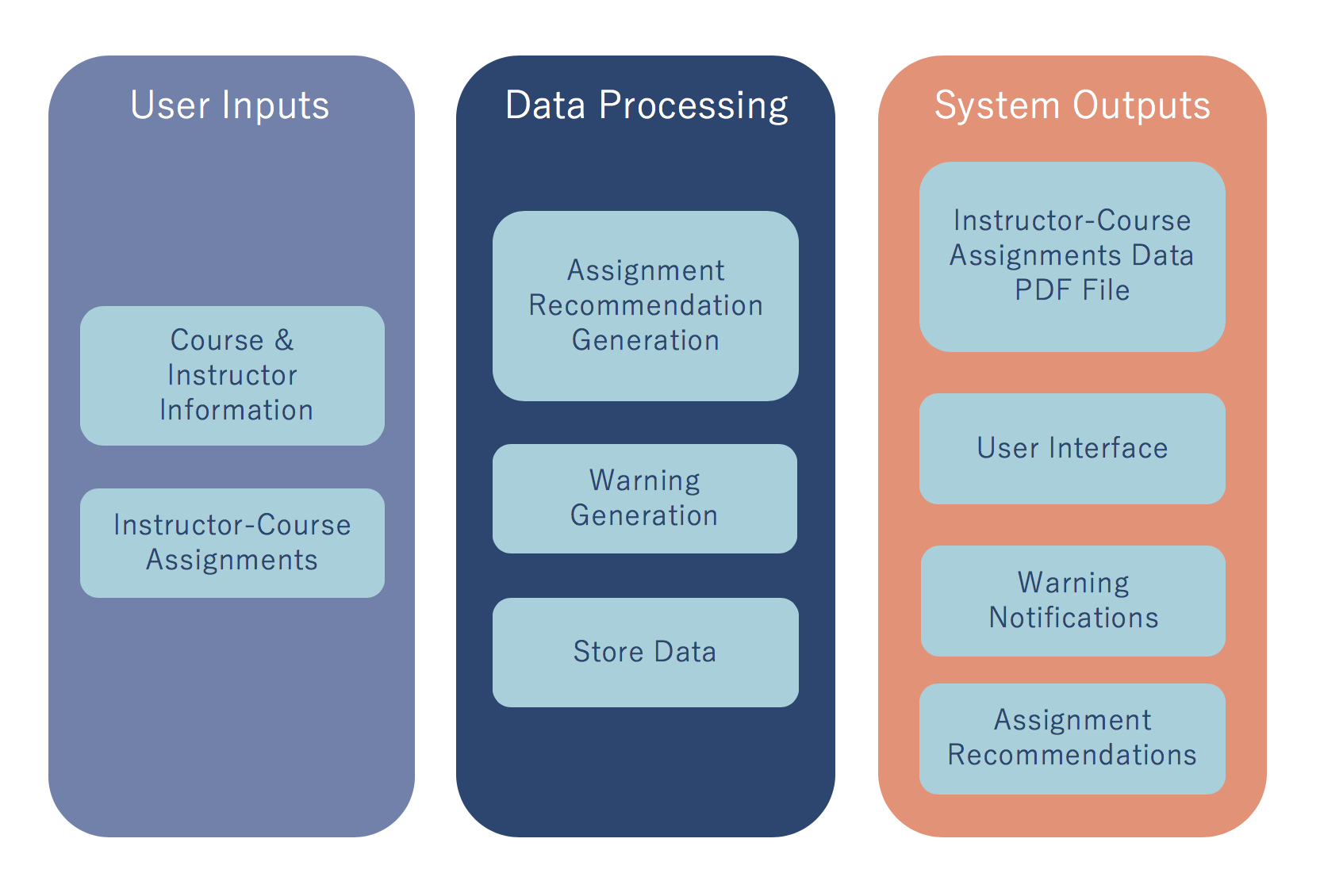
# List of Figures

# List of Tables

# Problem Description

Every semester at Texas State University, the program coordinators have the task of assigning instructors to courses and selecting a room and time for that instructor to teach that course. This task is simple in concept but becomes much harder to carry out when you add in all of the relevant information such as instructor’s preferences for specific courses, instructor’s availability and preferred workload, room availability and other variables.

Currently, Program Coordinators use an outdated, hard-to-navigate Excel Spreadsheet to assign instructors to courses. At the time, the Excel Spreadsheet, though inefficient, was the only way to manage and assign instructors to courses. Our project deliverable is an intuitive, user-friendly web-based application that provides a clear process of instructor-course assignment.



**Figure 1** Input-Output Chart –displays the inputs and outputs of our system.

It should be noted, the entirety of our system was designed and implemented by our team.

*The problem description section should tell the reader the topic your project is addressing and your specific deliverable(s).*

**BE REALLY CLEAR ABOUT WHAT YOU DESIGNED/BUILT.**

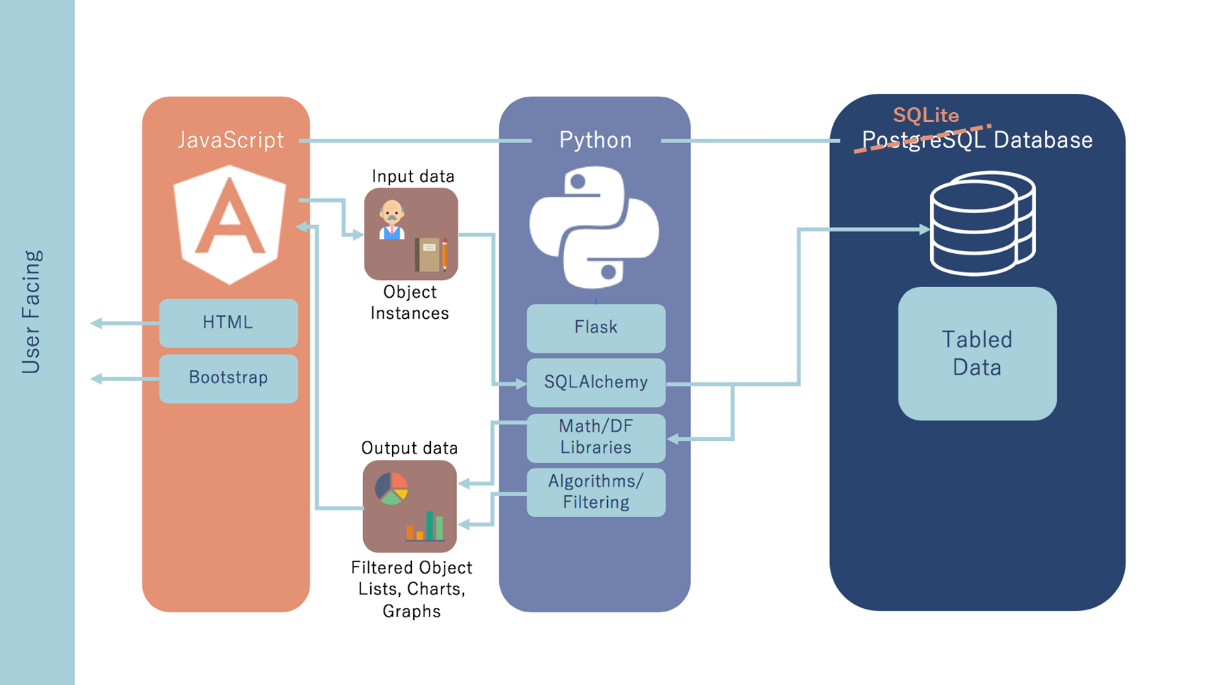
*Include a system-level diagram that will orient a reader to your design approach. The system level diagram should be very general. It should, at a glance, make clear what your project does, how it fits together, etc. It shall be Figure 1. Your work shall be highlighted in yellow as shown. Here's a couple examples:* (THEY ARE NOT CORRECT!!)

**HINT: Use block diagrams from previous presentations!**

# Progress Towards A Solution

## Design Decisions

Figure 2 illustrates our system design and the primary design decisions made throughout it. All of our design decisions are guided by the desire to create an ergonomic, efficient, modular system. In order to make the system modular we decided to separate the data-processing from the front-end display. Angular was used to generate the webpage since allows us to create a live application, and it handles the variable data from the server well.(sentence about other options). Python was chosen to implement the bulk of our data processing; this decision was made since it is a relatively simple language to learn and includes support for several interfaces for potential future work. Flask was used to implement the Python code since it provides a RESTful API, meaning that the program is stateless and pushes the bulk of the application data to the client-side application, allowing for less refreshes and a significantly reduced amount of data transfer. The database solution we selected was SQLitedb, originally, we were going to use PostgreSQL, however the scope was small enough that SQLite was more efficient while still providing the versatility and low time-complexity of SQL. The usage of a SQL database as opposed to a text file or other data solution allowed for concurrent accesses which allows simultaneous users.



**Figure 2** Software Architecture Design – the software design of our system, technologies used, and relationships between them.

## Design Approach

Our approach to the design modelled a top-down approach initially, by first determining what overall system architecture we would need for the basic deliverables, then what specific software interfaces and systems we would have to implement and finally the code itself.

Figure 2 shows the software tools used in the project itself, but in the design process we had to use additional tools to set up our design environment. We used the Ubuntu distribution of linux to set up a server to host our software, we also used npm and pip (package manager software for linux) to acquire the different software we would need for the project.

## Project Approach

We settled on the specific software used for the project early on. Once we decided to segment the project into discrete parts, we chose to have each engineer focus on a separate part of the project, specifically the front-end design, the data-handling and logical scripts, and the software framework and interconnection. The progress in each part went on apart from the other two, although in general each engineer was working on the same functionality, for example, the webpage to handle creating a new instructor was created when the python script to handle generating a new instructor in the database based on JSON data was being written.

Once the initial web-page was created, we establish a set format of JSON data to be sent back and forth between Angular (HTML/JavaScript) and Flask (Python), this allowed us to work on each primary function apart from the others.

## Engineering Standards

|  |  |  |  |
| --- | --- | --- | --- |
| **Standard** | **Title** | **Application** | **Relevance** |
| ISO/IEC 9075 | Information technology – Database languages – SQL | SQLitedb uses the SQL standard, as do all of our queries to our SQLite system | Data storage |
| RFC 8259 | The JavaScript Object Notation (JSON) Data Interchange Format | We use JSON to transfer data between the Angular frontend and Flask backend | Data transfer |

## Progress Towards Goals

Once we settled on the design for our project, the process of getting from start to finish for our deliverables was pretty cut and dry, although there were a few bumps. We settled on using Angular, Flask, and PostgreSQL shortly after starting the project, and were able to make decent headway in developing the front-end interface for the project. Once the front-end was developed, we started working on the different functions of the project, adding and editing instructors and courses, as well as displaying the lists of them. By the time these were finished, we realized that it would be easier to transition to SQLite, and store our database in a single file rather than trying to interface with a third system. The next steps were in developing the rest of the functions such as assignments, assignment recommendations, and warning algorithms.

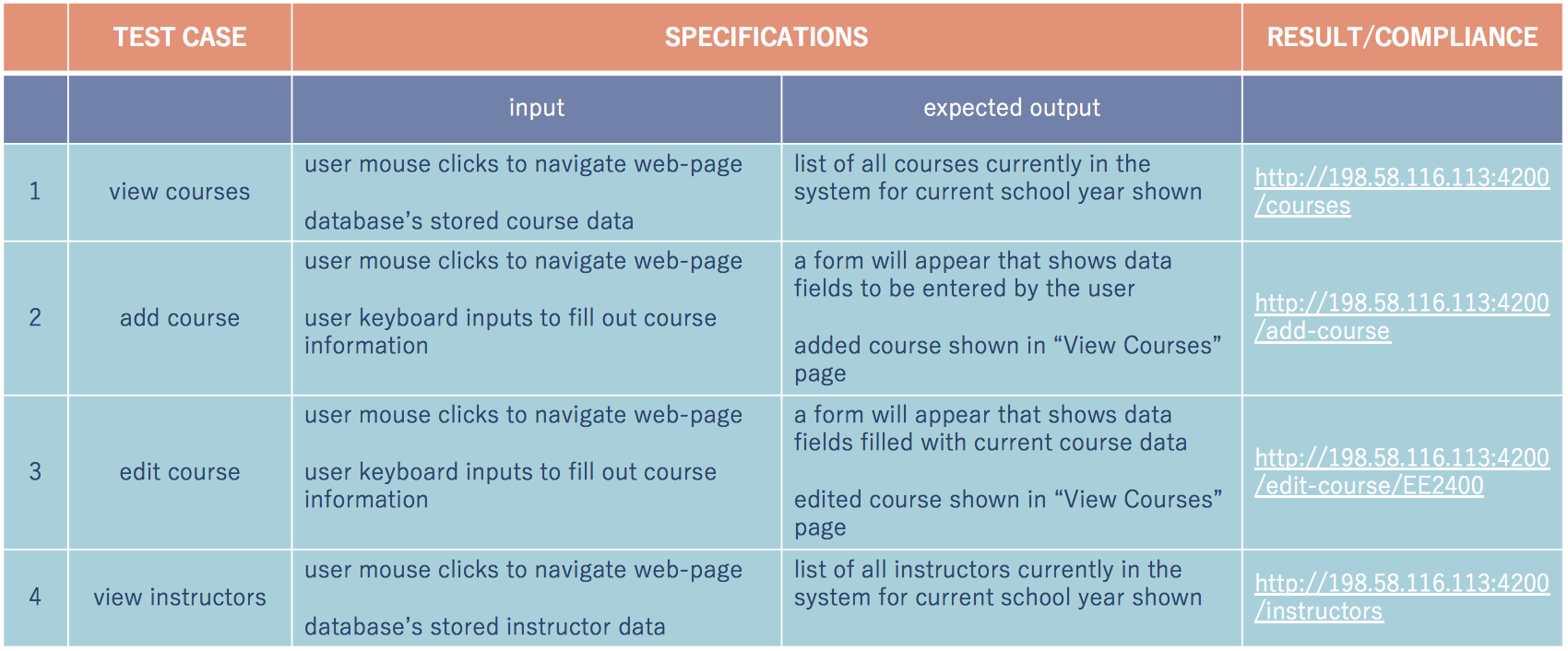
At the beginning of the project, our sponsor wanted us to interface with a software called Unitime which would be able to handle the class scheduling once our software made instructor-course assignments. As we worked through the instructor-course assignments, we realized that our software would not be able to interface with Unitime, this was an oversight on our part, as we did not realize that Unitime changed its formatting with every update. We looked at Unitime when our sponsor initially requested it early in the project, and at the time the format that Unitime used would work with our system, but some time during the project, Unitime was updated, and we realized that if Unitime updates its file format and how it takes in data frequently, then our software would have to be constantly updated to work with it. Under the advice of our sponsor, Unitime was then removed from the project since the goal of this project was to be stable without a need for updates.

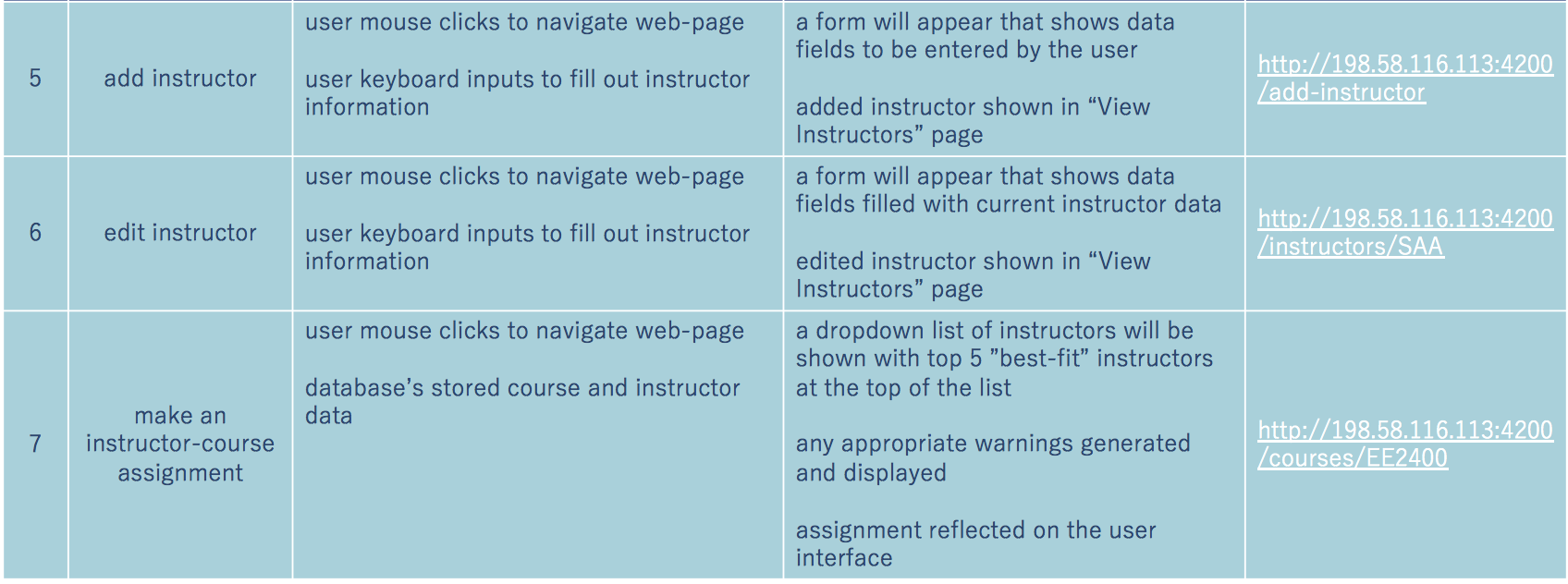
Another issue we ran into in the second semester was the login system. Ideally, we would have our software interface with Texas State University’s login system, since our software would be deployed on Texas State University Server’s, to be used by Texas State University faculty. When we contacted ITAC (the university’s IT department), we were informed that the entire project would have to be finished before they could begin to evaluate it and offer login security for it. Since the evaluation process could take several months after our project was finished, we developed a temporary username+key login system to be used in its place.

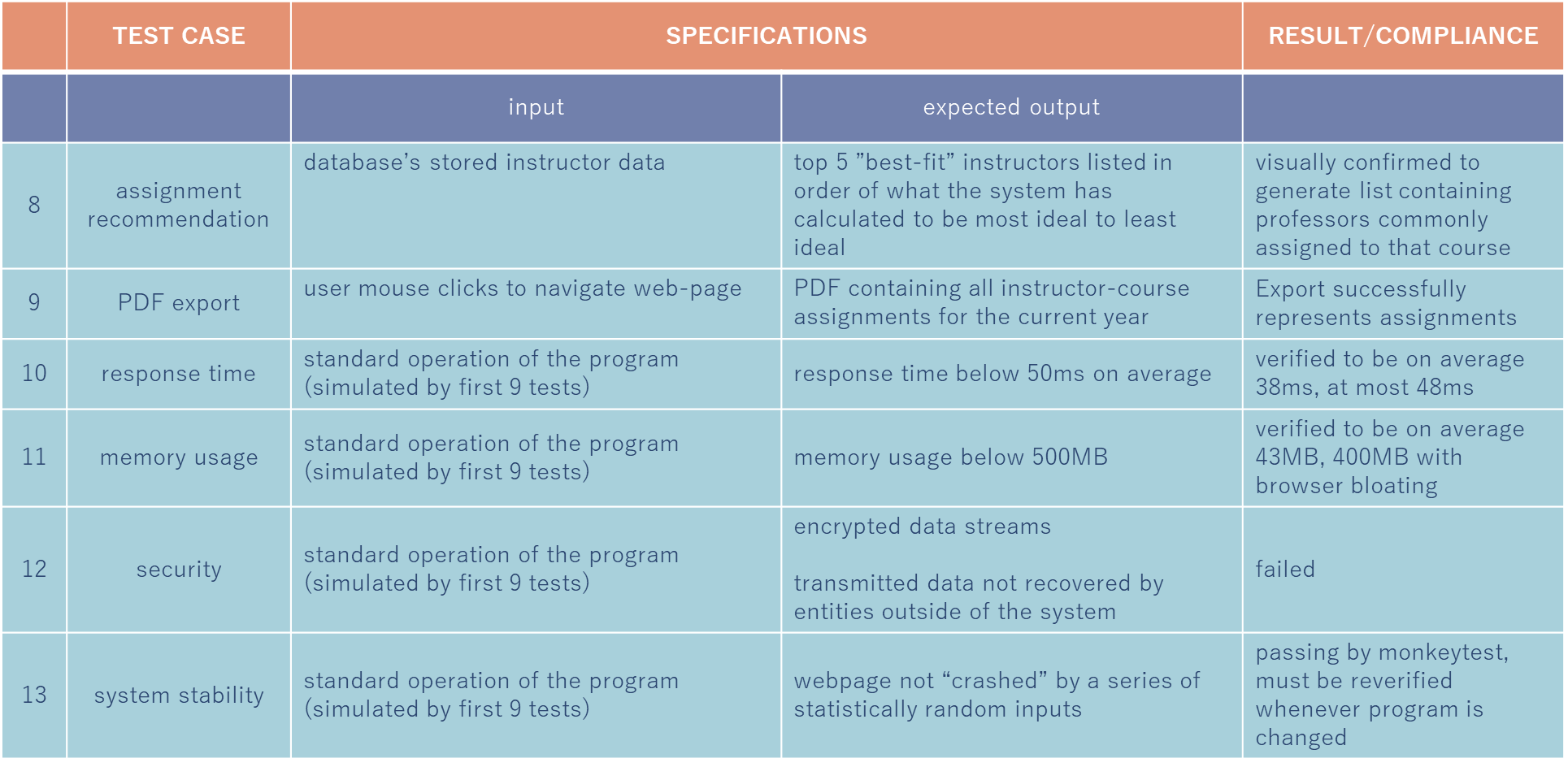
## Verification

Because of the nature of our project, it was difficult to verify the functionality of the program, the vast majority of our functions are either pass or fail, and don’t have a quantitative measure of success. The response time and memory usage were both verified throughout the development process with the assistance of Google’s Chrome developer tools.Tests 1-7 shown in figure 3 were all tested and verified visually throughout the design process, the entire database being populated multiple times by hand with the add courses, and add instructors functions, then verified by the view instructors and courses functions. The assignment recommendation was tested by setting instructor-course assignments to match a partial set of assignments from a previous semester, then verified by observing that the instructor who was actually assigned to that course was one of the recommended instructors. The security would be verified by using a middleman application that would try to decrypt packets, however since we haven’t been able to implement security measures, the tests were not performed.

## Characterization Results

**

**

**

**Figure 3** Test Cases

## Deficiencies

|  |  |  |  |
| --- | --- | --- | --- |
| Deficiency | Effect | Solution | Time need to solve |
| encryption | Data sent from server to client is not secured and could be intercepted if someone were so inclined | The easiest solution would be to program encryption and decryption on the endpoints of the system, at the point where python and javascript create and receive JSON packets. | It would likely take about 2 weeks to design and implement, then an additional 3 weeks to debug and fully integrate.; |
| login system | Login system can be easily broken into, allowing someone to corrupt the database if they had access to the webpage | The best solution would be for the program to have Texas State University’s login system embedded into it. | It would take anywhere from 2 weeks to 3 months for ITAC to evaluate the program. |
| class scheduler | Overall project is incomplete (although our scope is finished), since program coordinators still need to perform additional work outside our program to complete their job. | The system is designed to be modular, so the overall design can be expanded to include class-rooms, times, and student count | Based on the time it took to develop the instructor-course assignments, and the increased complexity of class scheduling, it would likely take about 3-4 months to fully develop the class scheduling functionality. |

## Iterations and Redefinitions

Our original project definition as outlined in our statement of work is:  
“Our product is an intuitive, web-based application that allows University Program Coordinators to assign professors to courses and export this data as an XML file. Currently, Program Coordinators use an outdated, hard-to-navigate Excel Spreadsheet to assign professors to courses. This method is inefficient, as it causes users to spend more time figuring out where features and data are located, rather than simply completing the task. Our product will benefit the sponsor by having a user-friendly, easy-to-navigate interface that provides a clear guidance on the process of professor-course assignment. By integrating the XML export feature into our product, Program Coordinators will be able to easily import this data into UniTime, a Class-Scheduling open-source software. Our product development team consists of three Undergraduate EE/CE students who are advised by a Sponsoring Faculty Member and two Student Mentors. We will be designing and developing the front end and back end of this web-based application using various programming languages and technologies. The project work will take place on and around Texas State University, San Marcos Campus. Our product will be ready in May 2019.”

One of the biggest changes in our design is the removal of UniTime as a requirement. As previously discussed, UniTime was determined to be unsuitable for our project, so its removal made sense. This change occurred halfway through the second semester, and didn’t negatively impact our project, since it was not integral to our design. The main lesson from this change was that in any design it is necessary to fully research any component of the design, and to avoid relying on external resources that you cannot ensure are consistent.

A big change between our original and current project definition is that we are much more specific about how we implemented the design, mostly due to refinement and completion of our design.

# Constraints

## Budgetary

*How did limited funds/supplies constrain your design?*

Our project consisted solely of software. All of the software technologies and resources used to implement coursebrew were open-source. The server used to develop coursebrew was already owned by one of our team members, Phillip Tran. No university funds were used to develop coursebrew.

## Design Feasibility

*You're not Intel so how did this constrain your design? How did equipment and software limitations constrain your design?*

## Manufacturability

*What constrains the ability of your design to be manufactured? What constraints did you consider?*

## Maintainability

*This is mostly for software, but hardware systems may require maintenance.*

The software architecture of coursebrew is designed to be modular. The database API is integrated into the frontend of our system via Angular Injectables. All of the frontend components are separated (home page, instructor, course, etc.) into their own modules and classes, which makes maintenance more accessible. The backend has maintenance functions so system administrators can manage the database outside of the frontend application. Developer documentation is provided for the frontend and backend of our system, along with directions to create an instance of coursebrew on a server.

## Environmental

*What environmental considerations did you have? Think about this in broad terms!*

Coursebrew is an application that is viewed and used completely via a computer screen. The outputs of our system are all viewed digitally. One feature of coursebrew is the ability to export a PDF containing all instructor-course assignments. If desired, the user may print a copy of this PDF, but it is not necessary and not a requirement of our system since the information can be displayed and shared digitally.

## Health and Safety

*What health and safety concerns did you consider as constraints?*

When designing coursebrew’s user interface, we considered the strain a user might feel on their eyes from looking at a computer screen. With this in mind, we designed coursebrew with soft colors and low flashing to reduce eye strain and avoid potential epileptic reactions.

## Social

*Speak to constraints due to intended users/audience, etc.*

The intended users of cousebrew are Program Coordinators, Program Directors, Administrative Assistance, and System Administrators. Access to the program itself will be restricted to these authorized individuals via a username and password combination unique to each user. The program is restricted to these users to avoid potential conflicts of interest or misunderstandings from other individuals involved in the University.

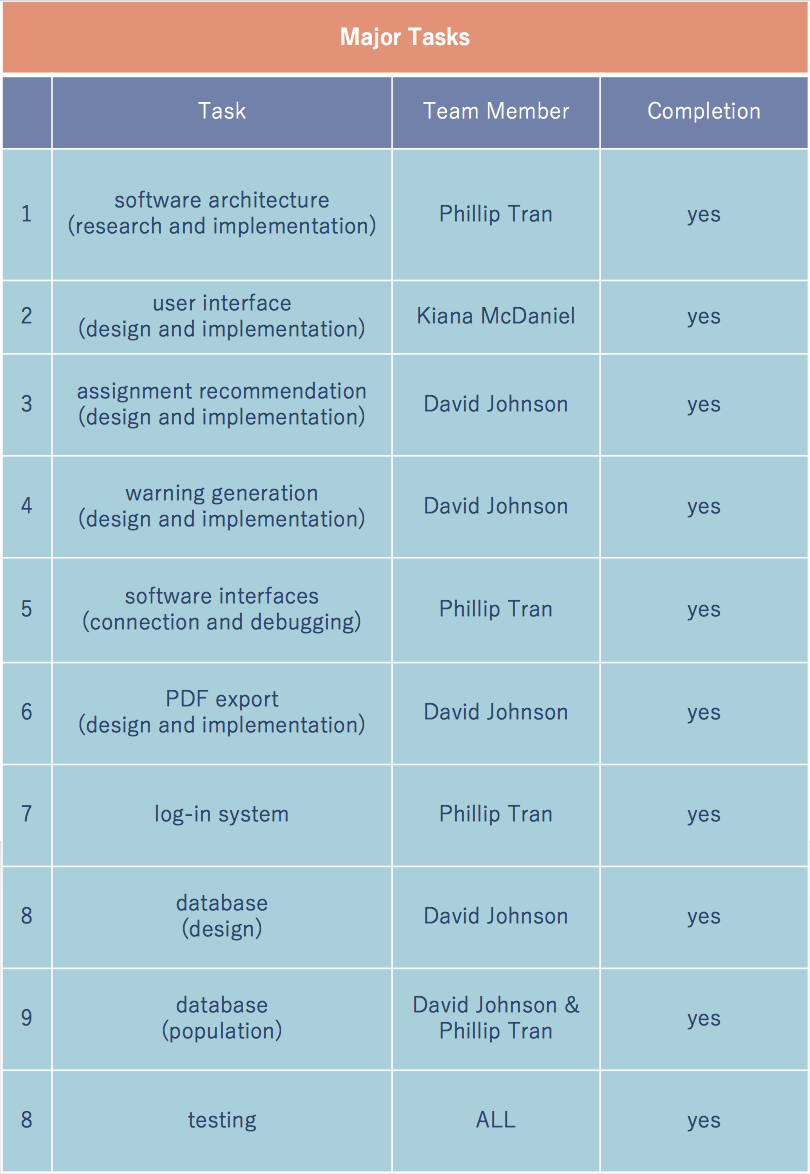
# Budgets

*The budget section should include a comparison of your proposed budget and the actual dollars spent to date. Create a table with side-by-side columns to convey this information.*

*Include a brief statement or paragraph summarizing your budgetary performance. Ignore items that differed by only a few dollars.*

# Work Schedule

*The schedule section must make clear to the readers which tasks were completed and which were not. Wherever possible make it clear which team member(s) were responsible for each* ***major*** *task. Discuss any timeline changes since the proposal was submitted. Use Gantt charting techniques (or side by side table if it’s preferable) to show the current status of the tasks in relation to the proposed schedule. However you do this it has to be readable!!! It will be a Figure or Table.*



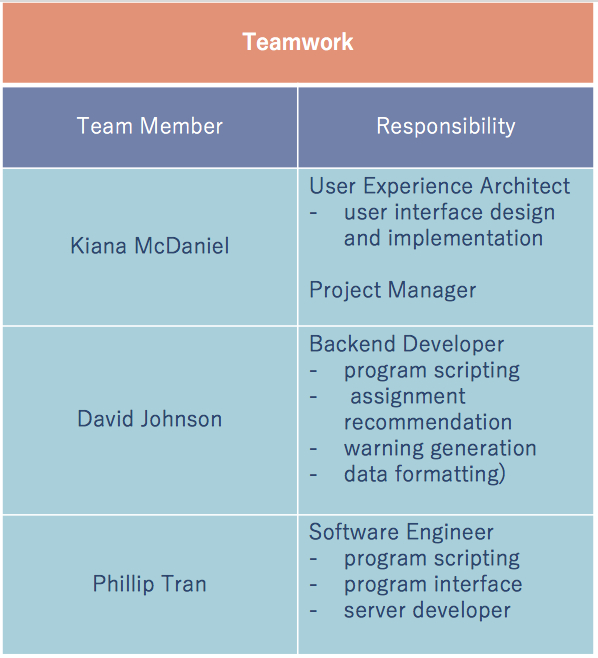
**Figure 4** List of Major Tasks

# Personnel Interactions

## Teamwork

*The teamwork section must clearly and concisely state the responsibilities of*

*each team member and his/her contribution to the senior design project. This is an expansion of Section 8. You can convey this information in a method of your choice, i.e. text, table, etc.*

**

**Figure 5** Team Members and Responsibilities

## Mentorship

## *What role did your Technical Mentor (Sponsor) and Faculty Advisor play? How much time did you spend with them and how frequently? How much did they assist you? What did they do? (point you towards resources, chalkboard lectures, help solve problems when stuck, etc) Be specific and give examples whenever possible.*

You're tiptoeing on political turf here so be very careful how you word this. If you got very little mentorship from your Sponsor/Tech Mentor then perhaps they did a great job defining the project, pointing you in the right direction, asking pointed questions, etc.

Our sponsor, Dr. C. Rich Compeau, played an important role in defining the requirements of this project, and provided excellent guidance and feedback to us throughout the entirety of this course. During our meetings, which occurred 1 to 2 times a month, we would update him with our progress and ask any questions we had about the requirements. He was very prompt in providing us with feedback and any resources we required. Dr. Compeau also pointed us towards Dr. Stan McClellan, who challenged us to think more about the scope of our project and helped us brainstorm solutions to key challenges dealing with instructor-course assignment. Our faculty advisor, Mr. Lee Hinkle, continuously motivated thinking within our team and gave us early insight on defining the hardware requirements in our documentation. He also pointed us towards Dr. Jill Seaman, who guided us on defining our software requirements for this project. We really appreciate all of the guidance we received throughout this course from all of the faculty members that advised us.

# Ethics

In working with Dr. Compeau on the instructor-course assignment project, the design team has become privy to particular information about the faculty at the university, and as per NSPE code of ethics 1.C we must “not reveal facts, data, or information without the prior consent of the client or employer except as authorized or required by law or this Code.” The majority of this information is in regard various faculty’s position in the university, their course preferences, and other information which should not be common knowledge among the student body at the university.

Since this project is entirely software based, there is little need for discussion of health and safety of its users, although the software has been designed with a soft color scheme and a lack of flashing lights to avoid eye fatigue and epileptic reactions.

# Summary & Conclusions

*Describe the overall capabilities and deficiencies of the system. (This is the more technical and detailed version, which you will summarize for the Executive Summary.) Provide a statement, based on the results of the system or module test, concerning the adequacy of the system or module to meet project requirements. How close did you come to your objectives?*

# Discussion

## Academic Preparation

Overall, our Texas State Electrical Engineering courses provided us with the critical thinking and problem-solving skills used to design and develop this project. Since this project is purely software, the hardware knowledge gained from our EE courses were not used. From our EE courses, we gained knowledge in designing and breaking down large systems into smaller, more discrete, manageable pieces. This aided us in designing and developing coursebrew to be modular, scalable and manageable.

All three team members have a concentration in Computer Engineering. This concentration includes completing multiple Computer Science courses. The knowledge gained in these courses helped us when defining data structures, test cases, and the object-oriented principles behind the software architecture of coursebrew.

Many of the skills needed to implement coursebrew were not provided through formal education. Our greatest resources throughout this project were Google and API documentation.

## Lessons Learned

*What did you learn about the engineering process? Teamwork? Management?*

As a team, we learned that the engineering process takes strong research, planning, and critical thinking skills. The ability to clearly organize and explain your thoughts is important in the design process. The ability to research and comprehend new skills is important in the implementation process. We learned that having the ability to work cohesively as a team is one of the strongest skills to possess. We learned to give and take criticism with a light-heart.

## Soft Skills

Presentation design and speech.

*What soft skills did you develop, improve or learn that you did not have before taking Senior Design? What elements of the course, or activities or assignments facilitated this learning?*

## Schedule Deviations

*What caused any deviation? What could you have done to better stay on track? What elements were under your control? Out of your control?*

## Staffing

It was adequately staffed, all of us were computer engineers which worked out great, any more engineers and there would be too much work done just coordinating our efforts, any less and we would be overwhelmed.

*Was your project adequately/correctly staffed? Why or why not? Enough members? Right major/tracks?*

## Final Observations

Start with taking everything into account, years, labs, etc. Trying to add it in later was bad.

Fully looked into the interfaces we were planning on using, we knew Unitime and TXState would be difficult to work with, but if we had looked into it closer initially we would have scrapped those requirements from the beginning and designed around a more realistic model.

*If you had this project to do over again, what would you have done differently?*

# Acknowledgments

Dr. C, Dr. McC, Mr. Hinkle, Dr. Seaman, stackoverflow

*Briefly acknowledge the individuals who helped you technically, organizationally, etc.*

*At a minimum you must acknowledge your Sponsor and your Faculty Advisor. Be generous!*

# References

*List relevant references. This section provides a bibliography of key project references and deliverables. This should not be a long section, but should show that you referenced and followed applicable guidelines.*