ENDEAVOUR

STUDIOS

S5 - Project Evaluation

SENG 321 - Group 4
April 12, 2012

Executive Summary

- The design process was useful, especially the enforced progression from S1 (overall vision) to S2a (functionality) to S2c (implementation).
- Group meeting time was adequate to have useful discussions about the project.
- Limited experience with and differing interpretations of PDOA would have made attempts to use it difficult. CLAWS lends itself well to the simple workpieces, controlled behaviour, and information display problem frames.
- All of the requested appointments functionality was implemented successfully.
- All of the requested scheduling functionality was implemented successfully, with the exception of some advanced UI portions.
- All of the requested Cat-a-logue functionality was implemented successfully.
- All of the requested users functionality was implemented successfully, with the exception of password recovery.
- Future improvements would include more robust error checking and tamper prevention, and more consistent UI styling.
- The only consistent roles were the team leader and webmaster, though this ended up working well.
- Management plans were accurate for documents, but much too optimistic for development tasks.
- The testing plan was also too optimistic; development lasted so long that little time was left for formal testing.
- Development tasks would have gone smoother if members had started becoming familiar with web development earlier. Teaching sessions hosted by more experienced group members would have been a large help.
- In a real project, more effort would be put into keeping to the management plan, and more testing would have been conducted.

Table of Contents

1.0 Design

1.1 Process

The design process provided a useful transition from one stage to the next, and enforced a clear, useful order. The progression from S1 to S2a to S2c was important for the group to decide on the general concept, then the specific functionality, and finally the specific implementation. These documents contained much overlap and repetition, but this was somewhat helpful in clearly establishing the function of the system. Lab sessions provided enough time to have useful group meetings and arguments on a variety of subjects, which was a pleasant change from similar group projects.

The design process put too much emphasis on diagrams, whether from UML or PDOA, and many groups ended up giving complex representations to bewildered customers. As the customer will likely have little understanding or use for many of the diagrams, especially class diagram, these should be saved for S2c.

1.2 PDOA

If the group had attempted to use PDOA in this project, efforts would have likely been bogged down by differing interpretations and uncertainty. It is difficult to predict how the final system would have been changed by the use of PDOA, but some problem frames can be identified in retrospect.

CLAWS was centered around the management of appointments and shifts, which could be edited by specific user types, and was therefore most obviously suited to the simple workpieces frame. The frame concern of dealing with invalid or nonsensical input would have been largely avoided because the group had full control over the user interface; only permitted actions were shown, and illegal actions were immediately met with 403 or 404 error codes. The information display and controlled behaviour frame could also be applied, though everything in the system was an abstraction and few parts of the physical world were actually being monitored or controlled. The Catcams were a simple example of both the information display and controlled behaviour problem frames.

2.0 Development

2.1 Implemented Features

2.1.1 Appointments

For the design of the appointments section, the majority of specified features were implemented. These features included the ability to schedule appointments of both grooming and kennelling types, with dynamic calendars that populate based on the availability in the schedule. When scheduling a grooming appointment the calendar will not allow times to be selected when there are not enough employees available, and likewise, the kennelling calendar will not allow selection of dates where all the kennels are full.

Alongside of appointment creation, editing and deletion were fully implemented. A employee, manager, or customer that owns an appointment has the ability to edit all the details with the exception of the appointment type. An employee or manager also has the ability to assign a cat cam to a kennelling appointment if needed, allowing the customer to view the specified cam. The appointment list was implemented in two views, depending on if the user was a customer or employee/manager. A customer is provided with a full list of all their appointments, allowing them to view historical or upcoming appointments as needed. The employee view provides a calendar style layout of the appointment list, allowing an employee to see exactly when appointments are happening and to prepare accordingly.

2.1.2 Scheduling

The majority of specified features were implemented for scheduling as well. Employees could view the schedule, including details of specific shifts, but could not change the schedule in any way. Managers could create, edit, and delete shifts as necessary, and changes were visible immediately on the shift calendar. One of the less important features, the ability to calculate the number of hours an employee worked in a given date range, was also implemented. As mentioned previously, scheduling was tied in to appointments; the number of employees on duty in a specific appointment slot limited the amount of appointments that could be scheduled for that timeslot.

While scheduling was fully functional, the shift calendar differed from the planned UI. Showing shifts as vertical bars proved to be too ambitious given the group’s limited web development experience; instead, the IDs of employees working during each hour slot were shown.

2.1.3 Cat-a-logue

The Cat-a-logue was fully implemented as specified, with slight improvements. All users could view the Cat-a-logue. Product details were shown on mouse-over, and clicking on a product brought up further information, though in a dialogue instead of on a new page. Using a dialogue eliminated the need to load a new page, and a full page for only one product would have looked quite empty. Employees and managers could add new items by filling out a form; they could also verify that the image path they entered was typed correctly before submitting. They could also delete items through a button on each item.

2.1.4 Users

The majority of specified features were implemented for users. CLAWS offered no way to add new employee or manager accounts, as there should only be one of each, but customers could register with an email address and a password. Customer accounts did not become active until activated by an employee; employees could also deny new customer accounts to delete them from the system. Customer could use their active accounts to log in and book appointments, and could change their account information. Employee could change the information associated with any customer account, and managers could edit any account. All pages were user-sensitive; certain buttons or views were added or removed depending on the user type.

Password recovery was not implemented, due to the work involved in configuring a mail server and the limited development time.

2.2 Future Improvements

The majority of unimplemented features involve error checking and input validation. When creating and editing appointments the client-side will not allow form submission with errant data (such as conflicting or overlapping dates/times for an appointment), however a malicious user could easily submit data that is not correct, throwing the system into an inconsistent state. The problem could be remedied by using more strict data validation when processing inputs, before submitting changes to the database.

Other improvements that could be made involve certain display and style changes. There are some inconsistencies with the style of the date/time selection calendars varying from the grooming to kennelling forms, and these could be remedied to make the experience for the user more concise. Improvements could also be applied to the appointment calendar, making the duration and display of appointments much more clean and well styled, to better match the theme of the site itself.

3.0 Management

3.1 Structure

Our management structure was defined at the very beginning of the course, and after it was defined, became somewhat less useful because some of the roles were not required until later in the project. The roles that did seem to work were the manager role and the webmaster role, simply because they were constant throughout the project. The manager role was handled well by Jared, and he did a good job ensuring that we always had the documents ready in time, and always gave good forward direction to the team. The webmaster role was done by Braden; this role was necessary since at the beginning, since we had to build a site for our company, and update the documents on it whenever we finished them. However, other than these two roles, all of the people in the team were focused on just writing the documentation and describing pieces of the system, regardless of what their actual role was.

Once the project reached a stage of development that required thinking about implementation details, then the roles started to develop further. The webmaster continued and created a theme and general layout, based on the prototypes and the others continued as web developers. After the theme was built, the developers were split depending on whichever section they wrote about in the documents. Instead of having roles like documentation expert or toolsmith, we had everyone do documentation, then everyone coding.

3.2 Plan

Assignment of tasks was usually successful. The group proved effective at dividing work equally. Setting individual deadlines worked well for document sections, but not so much for development tasks. Though the group was accurate in predicting which tasks would take the longest, almost all development tasks were underestimated, some massively. Too little time was allocated for UI cleanup and styling. Deadlines were too optimistic, and some dependencies between tasks were not anticipated. Dividing the work equally by implemented functionality did not work because of the large differences in group member experience with web development. It proved difficult to meet the tight schedule, and development stretched out until the day before the demo, though everything was completed to satisfaction.

4.0 Testing

4.1 Plan

The testing plan, depending heavily on the management plan, was also too optimistic. With so much development to accomplish in such a short time, the majority of testing accomplished was simple functional testing by developers. There was not enough time to create, let alone learn how to create, unit tests. Most integration and acceptance tests were very informal, though they did prove useful. Theo, who was not involved in development, was able to run a set of formal tests he devised, and reported the results to the group.

5.0 Lessons Learned

The CLAWS project was a very large success, due largely to the fact that we started coding and designing the system in parallel with the previous two deliverables. This was the key to the success of our project. This great time management should not be considered as the approach to get the best possible results though.

In an ideal situation with more time we would have taken the time to have everyone learn more about CSS, Javascript, and HTML so that it would not have so heavily been placed upon the ones who did know it. Armed with this extra knowledge, all the developers could have been easily put to task with much less help from the web developers which would have resulted in more detailed features, more error checking, and maybe an email system.

5.1 Understanding Web Development

At the beginning of the development phase of our project it was a constant splitting of time of the web developers to help all others with the ability to link their code to the web pages and create appropriate HTML pages and javascript calls. This need somewhat subsided as the project went on as it became easier and easier to find sample code elsewhere within the system to model new calls or pages but this code would not always be fully understood which would lead to new errors.

To remedy the situation if we were to start again, we would maybe sit down for one or two sessions to learn some of the basic concepts ourselves with maybe a little instruction from the web developers. This would have allowed each developer to have been more independent throughout the project most likely creating a better result or having been done the project faster which would have allowed for more testing.

5.2 Additional Error Checking

This is a problem that was introduced by the time constraints of the project. If we were to do this project again with the knowledge we have now we may have started the development a week or two earlier so that we could have considered more of the error cases and added appropriate checking to the functions. We believe that if we had had a little extra time to analyze errors and add the appropriate error checking that our system would be very close to ready for actual deployment minus the physical hardware.

5.3 Differences From a Real Project

Overall, the group's effort and commitment produced a product that is close to an actual, demonstrable, "real" project. That said, if working on a real project, more time would have been spent to iron out any discrepancies between group members. More meetings would have been scheduled, especially during the S1 time period, to make sure all members were on the same page about the product being created. More effort would also have been spent to keep to the development plan; the testing plan required meeting the development plan, and much more testing would have been necessary before presenting CLAWS as a real project. Development tools and frameworks would also have been selected from those the group was already familiar with, or research would have been conducted beforehand to determine the best alternative.