

Vision for library system

The system is supposed to be a library system, so the end vision must of course be a system which would help libraries in their day-to-day work of cataloguing, lending and retrieving books.

This system requires a few things for basic functionality from the client perspective. First of all, there needs to be a way of adding new books to the system, as well as modifying and deleting existing books. Each book needs to have a unique id to easily separate books from the same author or with identical titles, different printings of the same book or just multiple copies. Other than these, there need to be some fields to provide information about the nature of the book, such as genre. Since the system is, at least at this time, not aimed for commercial use, implementation of a standard library cataloguing system is not planned. A place for a description would however be helpful as well. The functions needed for this system to be working as a personal library is adding, removing, editing, and listing books both as a whole, and searching by author or title.

Current status

The status of the application as the project starts off is that the client side is finished and has an interface that will work, as well as an API for the server to work with. The server side is however non-functional, with only the very bare bone structure of code in place.

Stakeholder goals

The customer of a system like this wants a library system for personal use with the functionality stated above, within the deadline.

Developer goals: Meet deadline, with functionality in place and a robust code base, tests and documentation that would allow for refactoring or expanding the system later. Also, pass the course.

Tasks

In each iteration of the project:

- Plan the project
- Design the project
- Plan and write tests
- Implement functionality
- Make sure everything works

Project plan for library system

Introduction

The project is to build the backend for a working library system from a framework supplied by the course managers. The work started in the end of January 2018, and the project is due March 18th 2018, although there will be a fall-back option later on during the spring in case the project plan fails and the time-frame needs to be extended.

Purpose

The primary purpose of the project is primarily to learn the methodology for developing software, with focus on the planning, testing and documentation parts. The secondary purpose of the project is to build a working library system for personal use.

Stakeholders

The primary stakeholder in this project is the developer - the student working on the project, i.e. me. This stakeholder is affected by the success of the project in whether or not I pass the course, and eventually whether I get my teacher's license in Interface Design - which is the reason I'm doing this.

Other important stakeholders in this projects are the teaching staff for the course, being affected in the way that they want as many students to succeed as possible, with good results, and want their project to be perceived as educational, challenging and rewarding.

If this was a real library system project, the stakeholders would also be the customers asking for the system.

Scope

The scope of this project is limited to developing a working library system aimed at personal use rather than commercial, given that the functionality is limited to adding, editing, removing and listing books without adhering to any existing library conventions. The front end in this system is already provided, and so the project is limited to coding the back end. The time available is limited to January 15th 2018 - March 18th 2018.

Resources

Resources required for this project:

- Lectures provided by LNU for theoretical background
- Client side code provided by LNU, as well as skeleton for back end.
- Software Engineering 10th ed. by Ian Sommerville
- Project instructions on github

Project Goals

- Finish the project within time allotted
- Have a working product with the ability to add, delete, edit, list and search for books by title and author.
- Finish the project with enough quality to get a passing grade in the course.

Risk analysis

Risk	Probability	Effects
Crisis at work means I won't have time for studies	Moderate	Catastrophic
Crisis in own life means I won't be able to complete project	Low	Catastrophic
Illness means I miss study time	Low	Serious
Time allotted for study isn't enough to finish project	Moderate	Serious
Complexity of project is underestimated	Moderate	Tolerable
Complexity of other course demands I take study time from this project	Moderate	Tolerable
Computer breaks down	Low	Tolerable

Overall, the largest risks that I can see to me failing to complete this project is that my time will have to be devoted to my job instead of my studies - working full time as Lead Teacher at a school I've helped found means I tend to be the one having to jump in whenever something goes awry, and working with people, that tends to happen quite frequently. However, this time of year tends to be calmer than most, so hopefully I'll be able to only work full time and not more.

The other risks are that I have underestimated the complexity, or overestimated the time I can give to this project, but I count those as more moderate risks, since they can be mitigated by reevaluating and reworking the plan.

Work breakdown

Assignment 1

- Task 1: Personal planning
 1. Subtask A: Books
 2. Subtask B: JSON
 3. Subtask C: Web
- Task 2: Vision
- Task 3: Project plan
- Submit assignment for review by 2018-02-04

Iteration goal

Have an understanding of what the product is and what the specific goals of the project are.

Assignment 2

- Task 1: Analysis
 1. Subtask A: Use cases
 2. Subtask B: Diagrams
 3. Subtask C: Use case realization
- Task 2: Design
- Task 3: Implementation
- Submit assignment for review by 2018-02-18

Iteration goal

Have an understanding of system analysis and design, and design documents for a few of the functions to be implemented.

Assignment 3

- Task 1: Test plan
- Task 2: Test cases
- Task 3: Unit tests
- Task 4: API tests
- Submit assignment for review by 2018-03-04

Iteration goal

Have a plan for the testing of the product, and the first few tests in place.

Assignment 4

- Task 1: Planning

Iteration goal

Have a plan for the different iterations of the last stage of development, including the plan, design, testing and implementing of each iteration.

- Task 2: Iteration #1

Iteration goal

Having designed, tested and implemented the search for specific book function, working with both title and author search options.

- Task 3: Iteration #2

Iteration goal

Having designed, tested and implemented the Add book function of the system.

- Iteration #3

Iteration goal

Having designed, tested and implemented the edit book function of the system, and tested that everything works from all previous iterations.

- Submit assignment for review by 2018-03-18

Project schedule

Assignment:Task	Start	Done
Assignment 1: Plan		
Task 1	2018-02-01	2018-02-02
Subtask A	2018-02-01	2018-02-01
Subtask B	2018-02-02 AM	2018-02-02 AM
Subtask C	2018-02-02 PM	2018-02-02 PM
Task 2	2018-02-01	2018-02-03
Task 3	2018-02-01	2018-02-04
Assignment 2: Design		
Task 1	2018-02-08	2018-02-09
Subtask A	2018-02-08	2018-02-08
Subtask B	2018-02-09 AM	2018-02-09 AM
Subtask C	2018-02-09 PM	2018-02-09 PM
Task 2	2018-02-08	2018-02-10
Task 3	2018-02-08	2018-02-17
Assignment 3: Testing		
Task 1	2018-02-20	2018-02-22
Task 2	2018-02-20	2018-02-24
Task 3	2018-02-20	2018-02-26
Task 4	2018-02-20	2018-02-28
Assignment 4: Project		
Task 1	2018-03-01	2018-03-05
Task 2	2018-03-05	2018-03-10

Task 3

2018-03-10

2018-03-18

February 1st - 4th: Assignment 1.1-1.3

Monitoring

Revise and update plan every Sunday.

Personal reflections on Project 1dv600

#Assignment 1

Task 1

Subtask A

The most complicated aspect of this task was really that I don't understand what the book object in the dao folder should be? In the end, I just made a file there and went on to create the book list in GetBooksResource. Again, I made the mistake of adding the book list to the callback to make it show on the website instead of console.log, so had to retrace and log it to the terminal instead. As always, the hardest part of any task is usually to actually understand what the task giver is saying/writing. This is something I try to remind myself of every day as a teacher, because it's often the most challenging task for both me and my students.

Subtask B

Converting to JSON is the work of a few minutes, so not much problem with the work this time.

Improvements: Firstly having a list of books in a function is really not a workable strategy for a site where you'll need to add new books through an interface, so first of all I need to separate my books from my functions, and then require that json file into my function. This is probably not good enough for the final version, but it's better than what's there now.

Secondly, I need to comment my code so that I know what I'm doing, and add more details to my books.

Subtask C

Really didn't need much time for this part, and I think maybe I should have come up with more complex improvements - however, without being quite sure what the requirements for the system is, I don't want to rush ahead with something major before I have implemented the basics thoroughly. I realise that my json file is very transitory, as there will be xml in the works, but still, it bugged me having a messy function.

Task 2

The difficulty with writing a vision is that I still have no clear idea of how complex this system can be during these few weeks, or whether to see as a library system for a full-fledged library, as a small scale version mostly used at home, or as a system for a smallish bookshop. They all have different demands. I went with the real library-angle, since I'm guessing that's what the basics of the system was intended for, and it may be difficult to work against what the course idea is. In reality it appeals more to build a system for my home or my school, since we don't really have one at the moment, but generally speaking it's better to avoid doing things you actually need in university assignments - they need to meet the requirements of the course first, and your own needs second.

Task 3

The project planning was mostly just a nice way of doing something with a bit more structure than I probably would have otherwise. Planning has been a crucial part of my life for many years now, juggling a complex work, studies and home life. The risk analysis was one part that I wouldn't have put into words had this not been part of the course literature, but it did help me focus on what the real risks are. This of course being a smallish project where I work alone rather than a large project with many people involved, it is a radically more simple plan than the ones I do work with for a living. Time wise I did overestimate the time it would take to do some parts of the first assignment, but the estimates were based on me usually being interrupted while I work, and this last two days I've had perfect calm to focus on this course!

#Assignment 2

##Task 1 ###Subtask A I have encountered UML-diagrams previously, and I do find them very useful in the planning stage for development. Use cases are of course very similar to ways that I've worked in Interaction Design, since I have a lot more experience from that field than programming, and so they do feel quite natural to have here as well. In general, I think the difficult part here is to decide on the scope of the planning stages of this project, knowing that a full scope library system would need more functionality than we can really make during a course like this.

Subtask B

Robustness diagrams in this very case seem a bit overkill for documentation and planning, but I can definitely see the usefulness of it to provide a link between the slightly abstract use cases to the actual implementation of the functionality. Especially for a larger scale product where the people implementing and the ones designing the system may not be the same people or even in the same location, this seems to provide a good basis for clear technical communication of how the different steps should work.

Subtask C

Sequence diagrams, again feel quite useful. But like with subtask B I'm starting to wonder how many different diagrams are really useful for this fairly limited project. For this kind of project, I do feel that I would do use cases and either sequence diagrams or robustness diagrams, but to add more would start to feel a bit redundant. Again though, I can see that the story is different for larger scale projects. After doing the sequence diagrams I feel I have a better grasp of the system architecture.

##Task 2 This way of doing things does really force me to think through things before I do the coding. Of course, it also means I can't test and see if I'm mistaken in some assumption of how things work - but then if I do find out in the end that I made a planning mistake I guess I can go back and redo the sequence diagram. It probably reduces the amount of errors in the end anyways. It is akin to what I tell my students about essay writing (my profession is language teaching), that is a process that is more about planning than actual writing if you want the end result to be of good quality and reduce the number of errors and unnecessary work you do.

Task 3

Ok so this one took a lot longer than I thought, possibly partly because I started it while I was at the

students' LAN party at work, and was quite tired and unfocused, but I really got tangled in how to transform the xml to json. It really should have been ten or twenty minutes work, but it took me two hours to figure that part out. The rest was really quite easy. Since the only real tangle here was the xml, I don't really feel that there was a huge difference between the designed and undesigned approach to implementing a function - the delete function was so much faster and easier to implement, because I had figured out the xml and understood the flow between different functions. I think the lessons I take away from this assignment is:

- What takes time is rarely what you think will take time - it's true about a lot of things in life.
- Converting information between formats is important, but can be time consuming (my partner who works as a bioinformatician says that this is mostly what working as a scientist is about).
- Planning the flow between functions and what information travels is important, but planning the overall architecture is more important than the details.

Assignment 3

Task 1

OK, so I'm not completely done, but I think this will work better for me if I come back to the plan and fill in some more details after I've done the first two tests and have a clearer view of what I should test. I also think I'll come back and replan the API-tests after I'm done with the Unit tests, instead of planning every test before I start.

Task 2

Wrote two fairly simple manual tests. I first thought of these as real user tests, but I realised after reading about it that this is not the case - coming from the field of linguistics, pedagogy, UX and behavioural sciences testing does mean something a bit different for me than it does in the field of development 😊. Anyway, I do think that manual testing is what comes most natural for me, since it tends to be the an integral part of my natural work flow, though I'm not very thorough in documenting it properly. I see the importance of doing more of these, and it feels a bit wrong to stop at just these two - but I'm as always pressed for time.

Task 3

This was really good - I have been meaning to start looking more closely at unit testing and testing frameworks, and mostly this was not too tricky, although I know I have done very simple tests thus far. This is definitely something I want to dig into further, for my own progression, but also because I know that this is one of the weaker areas at our school.

Task 4

The trickiest part of this was that I got a false positive until I figured out how to use the 'done' functionality when testing code with callbacks. This also made me realise I had a bug in one of the previously written tests. Good learning 😊 Other than this not much more to say about testing - it's kind of a vital part of any development, but that feels like a no-brainer (in theory, I know it's often lacking in practice).

Time Log

Assignment/Task	Expected time	Start	Finish	Actual time
Assignment 1				
Task 1				
Subtask A	45 minutes	2017-02-03 17.23	2017-02-03 18.16	53 minutes
Subtask B	15 minutes	2018-02-03 18.20	2018-02-03 18.30	10 minutes
Subtask C	60 minutes	2018-02-03 19.15	2018-02-03 19.55	40 minutes
Task 2	45 minutes	2018-02-04 14.22	2018-02-04 14.57	35 minutes
Task 3	75 minutes	2018-02-04 15.43	2018-02-04 16.37	54 minutes
Assignment 2				
Task 1				
Subtask A	120 minutes	2018-02-08 19.50	2018-02-08 21.46	116 minutes
Subtask B	45 minutes	2018-02-08 21.55	2018-02-08 22.30	35 minutes
Subtask C	60 minutes	2018-02-09 20.50	2018-02-09 21.29	39 minutes
Task 2	30 minutes	2018-02-09 21.51	2018-02-09 22.15	24 minutes
Task 3	120 minutes	2018-02-09 22.15	2018-02-09 23.22	
		2018-02-10 10.00	2018-02-10 11.20	
		2018-02-11 09.53	2018-02-11 11.15	229 minutes
Assignment 3				
Task	60 minutes	2018-02-22 20.51	2018-02-22 22.01	70 minutes
Task	60 minutes	2018-02-25 10.09	2018-02-25 10.37	28 minutes
Task	150 minutes	2018-02-25 10.40	2018-02-25 12.14	94 minutes
Task	120 minutes	2018-02-26 20.25	2018-02-26 21.45	80 minutes
Assignment 4				
Task				