MYPHARMA.IE/THÉIREP: IRELAND'S FIRST E-PRESCRIBING SYSTEM

Higher Diploma in Science in Computing

Word Count: 1178 (Body)

Higher Diploma in Science in Computing
Software Development

Contents

CIAN WALKER

10021419@mydbs.ie

26/09/2022

1. Introduction

Currently; there are two methods of prescribing and retrieving medicines in the Republic of Ireland:

1. A hard paper copy can be brought and retained (in the case of recurring prescriptions) to any given pharmacy.
2. General Practitioner surgeries can issue a prescription via email to one specific pharmacy.

This is a cumbersome process, but it is particularly inconvenient for those who have recurring prescriptions, who have to travel midweek for business regularly, or both.

In 2018, Ireland's Health Information and Quality Authority (HIQA) followed up on earlier research in 2012 (which emphasised a reduction in prescribing errors more so than convenience as a strategic goal) by outputting two reports.

The first report conducted an international review of ePrescribing adoption in Scandinavia, the Baltics, the United Kingdom, and the Benelux before expanding it's scope of inquiry to the USA, New Zealand, and Australia. It found that:

"Successful ePrescribing programmes typically balance local and national needs, continually sharing a clear national vision that also meets important local requirements. To ensure stakeholders' commitment to the programme, GPs, community pharmacists, vendors and others may need to participate more fully in designing and testing the service to ensure their needs are full met."

The second report generated six recommendational areas for the rollout of an ePrescribing infrastructure in Ireland and emphasised:

1. Roadmap and Legislative Requirements
2. Governance
3. Data Privacy
4. Stakeholder and Communication Strategy
5. Standards-Based Approach
6. Phased Implementation

Now that the COVID-19 pandemic is abating, the issue of developing an Irish ePrescribing solution is likely to tabled again imminently.

2. Project Scope and Objectives

There is a pressing need for a stable, cloud-based portal where prescription records can be written by General Practitioners (and edited/deleted) into a secure, cloud-based database; and retrieved by any licensed pharmacist in the country, and in a way that respects consumer/patient privacy. The various use case flows will be carried out by authenticated users of up to three different kinds: Physician and Pharmacist for certain, with potential use cases and functionalities for Consumer/Patient personas also.

- Portal & User Interface: Needs to be modern, interactive, and smooth with plenty of JavaScript to bring it up to modern UX standards.
- Users, OAuth, OpenID Connect: Initial User authentication (OIDC) carried out that should query Physician Irish Medical Council numbers as well as Pharmaceutical Society of Ireland before creating these identity records before authorising access requests (OAuth) against this directory. Again, consumer/patient flows to be developed in if project timescale and progress allows.
- CRUD APIs: The core code that orchestrates the operations behind these use cases. APIs that handle data transactions between the User Interface and the Database, whether the use cases be GP or Pharmacy registration, authentication, authorisation, or the submission and retrieval of ePrescriptions.
- Database: Must be relational, stable, and secure.
- Privacy: Interesting challenges around who should be able to view what details and for what reasons; particularly in the part of the flow where pharmacists are retrieving ePrescriptions.

3. Student's Learning Objectives & Technical Specification of the Project

I have combined the learning objectives and technical specifications sections of this proposal below since that way I can outline both my learning objectives and plans for the various layers of the application stack simultaneously.

I have built a very crude version of this application as part of my final Advanced Programming module. It utilised very Crude HTML, no JavaScript, SQLite3 database, and a set of Python Flask APIs to pass data back and forth between the Interface and Database. I will include the zip file of this entire project with this submission. I intend on using almost none of this technology save perhaps for one HTML template, and possibly not even this.

Because of my current role as an identity & access systems engineer (with considerable specialisation in consumer IAM), there are a number of skill areas I would like to use this project as an opportunity to develop:

-JavaScript and Node.JS: my original attempt at this app featured no JS and so looked very crude. NodeJS skills are in sharp demand in my industry, I am exploring whether Node is a viable environment within which to build this solution. One small doubt remains around it's DB integrability. MongoDB/NoSQL is what NodeJS natively works with as opposed to MySQL. JS is also an important extra skill for me to develop to enable me to build sleeker UIs.

-Significantly deeper API skills, particularly around OAuth/Open ID Connect flows. Ultimately, I am particularly attracted to application integration development so would like to upskill as much as possible here. NodeJS's API framework is Node-API.

-Database: SQLite3 is too lightweight for a real production application like this. If possible, I would like to explore Mongo/NoSQL usage in this project. There are also other ways to use combinations of NodeJS, JavaScript, with integrations with Python/MySQL if necessary. My research continues on this. Here is an draft schema (if SQL) of the data structure:
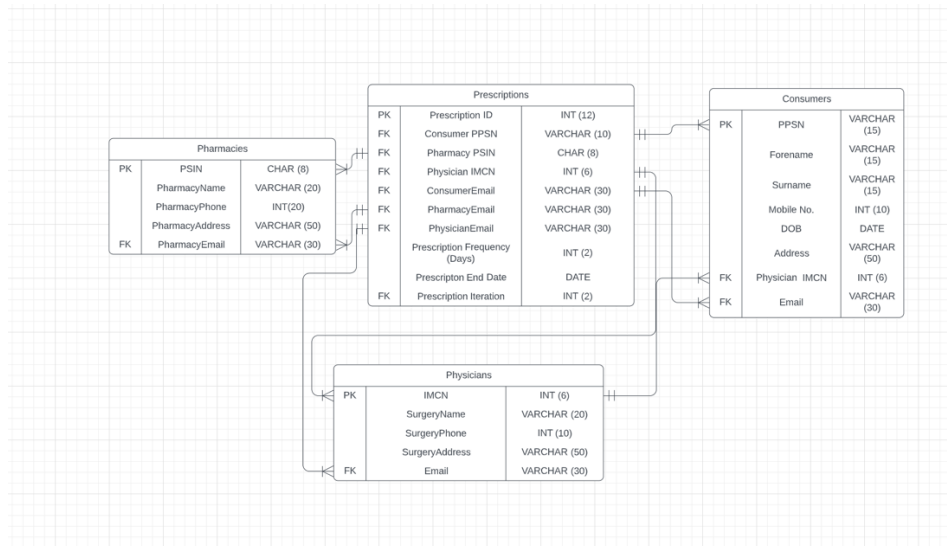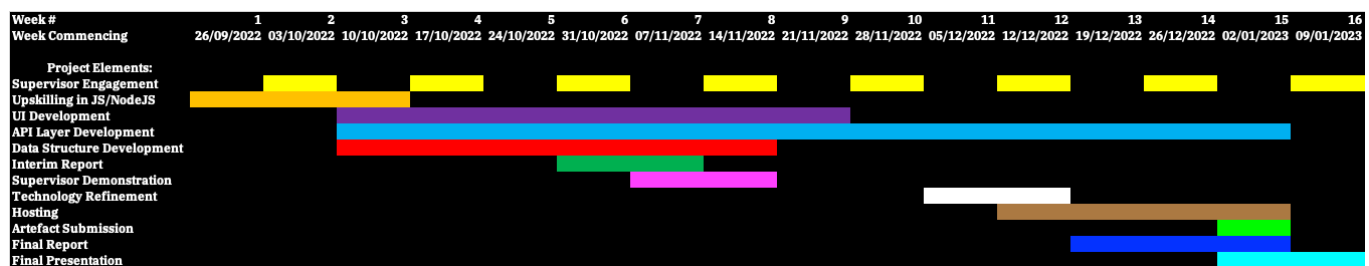

-Mobile Optimisation: dabbled very little in this but would aspire to outputting a mobile-optimised UI in addition to the web-app if possible.

-Hosting: Excited to execute on a proper, live production hosting of an application. I prefer AWS as a platform but Azure is a tempting option since my profession comes into contact with it more (I think perhaps it is more popular amongst UK & Irish enterprises relative to North America).

-Testing: There were a small number (two to three) of use case flows I failed to test properly in my last project. I need to be more mindful of testing use cases where users supply invalid data etc. and not make the mistake of only testing use case flows where users provide valid information.

In terms of development tools, I've been acquainting with Google Developer tools and also continuing on from my training with PyCharm (which may still be leveraged if the APIs end up needing to be written in Python). I have a student edition of all of JetBrain's other IDEs: WebStorm and IntelliJ will both be suitable for Node.JS development.

4. Project Plan

Week # Week Commencing	1 26/09/2022	2 03/10/2022	3 10/10/2022	4 17/10/2022	5 24/10/2022	6 31/10/2022	7 07/11/2022	8 14/11/2022	9 21/11/2022	10 28/11/2022	11 05/12/2022	12 12/12/2022	13 19/12/2022	14 26/12/2022	15 02/01/2023	16 09/01/2023
Project Elements:																
Supervisor Engagement		▓		▓		▓		▓		▓		▓		▓		▓
Upskilling in JS/NodeJS	▓	▓	▓													
UI Development			▓	▓	▓	▓	▓	▓	▓							
API Layer Development			▓	▓	▓	▓	▓	▓	▓	▓	▓	▓	▓	▓	▓	▓
Data Structure Development			▓	▓	▓	▓	▓	▓								
Interim Report						▓	▓									
Supervisor Demonstration							▓	▓								
Technology Refinement										▓	▓					
Hosting											▓	▓	▓	▓	▓	
Artefact Submission															▓	
Final Report												▓	▓	▓	▓	
Final Presentation																▓

I am envisaging the API development to proceed for the majority of the project period as it was the most difficult layer in my previous project. Data structure development is a strength of mine, so I forecast it will be the component within the stack I will have completed and finalised the soonest. I envisage this being followed by the UI, which in any case must evolve in tandem with the API development. Other elements to be scheduled with allocated lead-in times as portrayed in the above GANT.

5. Conclusion

In conclusion; due to the prescience of the need for this solution, I wish to pursue the idea used in my Advanced Programming project except totally re-engineered for Node.JS, with a profoundly slicker, JS-rich user interface. I'll attempt to develop the middle API layer using Node-API with Python-based APIs explored if time begins to elapse too quickly. I will either use a MySQL or NoSQL data structure. The only *potential* technology to be reused is the initial HTML template.

6. References / Bibliography

Health Information and Quality Authority (HIQA), 2013. *National Standard for Patient Discharge Summary Information*.

Health Information and Quality Authority (HIQA), 2018. *ePrescribing: An International Review*. [online] Available at: <https://www.hiqa.ie/sites/default/files/2018-05/ePrescribing-An-Intl-Review.pdf> [Accessed 22 September 2022].

Health Information and Quality Authority (HIQA), 2018. *Recommendations for the national, community-based ePrescribing programme in Ireland*. [online] Available at: <https://www.hiqa.ie/sites/default/files/2018-10/EPrescribing_Recommendations.pdf> [Accessed 22 September 2022].

Nodejs.org. 2022. *Node-API | Node.js v18.9.1 Documentation*. [online] Available at: <https://nodejs.org/api/n-api.html#node-api> [Accessed 26 September 2022].

W3schools.com. 2022. *Node.js MySQL*. [online] Available at: <https://www.w3schools.com/nodejs/nodejs_mysql.asp> [Accessed 26 September 2022].