SmartCare Surgery Case Study

1. Specifications

SmartCare Surgery is a GP business recruiting 1 full-time and 1-part-time doctors and 1 part-time nurse. A weekly basis working scheme is put in place requires 1 doctor and 1 nurse to serve every Monday and Friday while 2 doctors to serve the rest of the week.

SmartCare manager decides to replace their paper-based system with a web-based system on which doctors, nurse and patients can access to it via web for self-service. It is known that SmartCare Surgery works with NHS and serves privately, where the charges per consultation is billed based-on the length of consultations (each standard consultation slot is 10 minutes) and the issued invoices either sent to NHS or paid privately. All patients are prescribed or forwarded to hospitals for specialist doctors. The care and services provided by the nurse follows the same time frame, but, invoiced with a different rate.

The manager thinks that it would be wise to start with a prototype to make sure if the idea for such a project is feasible. For that reason, she communicates to SmartWare Co. software company to submit, demonstrate and champion a prototype solution.

It is expected that the system-to-be would serve for 3 types of users (doctor, nurse, patient) and 1 admin user. Doctors and nurse should be able to see all daily patients timetabled, to issue a prescription at the end of each consultation and surgery, while patients should be able to book an appointment and to request re-issue of a repeating prescription online. The system should produce an invoice following each surgery. An admin needs to take care of records, user operations and to produce all weekly documents such as turnover, private payments and charges sent out to NHS. The system should be able to calculate periodic turnovers, e.g. daily, weekly and monthly.

Access to the system will only be allowed through successful login, and each user should be logged out on request or automatically after a non-active while, e.g. 5 mins. The signup operation should be authorised by admin if it is for a doctor or nurse user, otherwise can be done automatically.

Booking a surgery appointment requires a calendar functionality to use, which can be invoked from an existing calendar service such as *Google Calendar API* as a ***web service***, or can be developed from scratch. Additionally, address lookup can be handled through another ***web service*** such as *Google Map Services*.

None of the functionalities must be accessible without user authentication and authorisation, which necessitates use of a substantial login systems. The system needs to restrict jump from one user's dashboard to another unless authenticated through authorisation process. SmartCare manager also thinks that they would be able to collaborate with some other GP services for exchanging patients, forwarding nurse operations to one another. An advance version of the system will please the manager if it allows this type of collaborative operations to be made available.

2. Expectations

The Web Application is expected to meet the following requirements:

1) At least the following pages are expected to be included:
   a) The main (home) page letting users select the type of user and action ahead
   b) Login/registration page for member users is required
   c) A Dashboard page for "doctor" users
   d) A Dashboard page for "nurse" users
   e) A Dashboard page for "patient" users
   f) A Dashboard page for "admin" users that lets to process the operations as required
2) A user should be able to navigate through the pages, smoothly, and especially be able to access to its own dashboard and the home page from any page.
3) A member should remain logged in until either the session is timeout or user is changed/logged out. With this respect, access to any different member's dashboard must be secured.
4) External services can be used for acquiring calendar services (such as Google Calendar or any other calendar service or API) to use for booking appointment with doctors or nurse. Also, another web service (e.g. Google Map services) can be used to look up users' addresses and retrieve them automatically.
5) Database is required to be separately deployed and connected via Docker containers. The work done for Advanced Databases module is allowed to be reused.
6) Since this group work is organised to collaborate with Advanced Artificial Intelligence (AAI) module, the work to be done for AAI module assessment can be deployed on another separate container and be accessed via web/restful services.
7) The whole system should be using:
   (i) Python Django following MVC (MVT) patterns,
   (ii) A Database system, (e.g., MySQL, PostgreSQL etc.,) should be separately deployed and accessed for handling data,
   (iii) Deployment should be conducted via Docker containers,
   (iv) Django Restful Framework (DRF) should be used together with Docker for creating, deploying, and accessing the services.