



GOODX SOFTWARE / QUANT ENGINEERING
SOLUTIONS

RESEARCH & DEVELOPMENT - Q4 2023

Software Development Applicant Test

API Booking System

Authors:

R.J. FOURIE

L. LOUBSER

Last revision: December 1, 2023

Contents

1	Introduction	2
2	Background	2
3	Prerequisites	2
4	Important notes and considerations	2
5	Assignment	3
5.1	Tasks	3
5.2	Context	3
5.3	Requirements	5
5.4	Recommendations	6
5.5	What's in the box? [1]	6
5.6	Resources	6
6	Submission	7

1 Introduction

The following is a technical proficiency test to assess the skill level of the applicant. **Please complete and submit the assignment within 2 weeks**, unless otherwise specified. Should you have any questions, please contact the sender of this document.

2 Background

GoodX was founded in 1985 by Dr Dirkie Wolvaardt based on years of experience, continuous research and development. This has culminated in a product that is used and trusted by practitioners and medical administrators all over Southern Africa.

The suite of GoodX medical administration and billing software was developed by a company filled with people who know and understand medical practice financial administration and management. GoodX has more than 195 employees and we take pride in having plenty of experience in the medical, financial and development fields. This combination allowed us to develop a product that is easy to use, secure, accurate and integrated. This, in turn, frees medical practitioners up to focus on their areas of speciality while our software facilitates the workflow in the office [2].

3 Prerequisites

As an applicant you must be familiar with the majority of the following topics in order to successfully complete this test:

- Familiarity with HTTP and general knowledge of APIs [3].
- HTML, CSS, Python and JavaScript.

4 Important notes and considerations

- During this test you will be interfacing with a public environment (API and database). These resources may be in use by multiple users causing the returned data from the API to be inconsistent.
- The environment can be subject to unscheduled updates and short periods of downtime.

5 Assignment

5.1 Tasks

Your task is to design and develop an application which can be used by medical staff (e.g. a doctor or nurse) to get a summary of their daily schedule (diary), as well as perform basic operations on their diary. Your application will need to interface with GoodX via an API that is part of the GoodX web application (referred to as GXWeb).

You may choose to develop either a desktop, web or mobile application using the language and/or framework of your choice. **Please note: When choosing a web application, you are also expected to deploy your solution to any hosting platform. (e.g. Google Firebase)**

Your application must include a basic Graphical User Interface (GUI) and should be able to perform basic CRUD [4] operations, related to Diary Bookings, by integrating with the given GoodX API. Please note, as mentioned in section 4, the provided API is an actual GoodX API used in production.

5.2 Context

In order to use the API you must authenticate yourself using the */api/session* path (using your provided username and password). Performing this call will return a session UID, to be stored in cookies, for future requests. This will allow you to make all the other calls with that session UID as authentication. A user account has access to different Entities, each entity has Diaries, each diary has Bookings. Bookings have types, and statuses. See the section below for more information.

Relationships of GXWeb, Entity, Diary, Bookings, etc.

- A GXWeb install can have multiple **Entities**. (Usually an entity for each medical practice in a group)
- An **Entity** can have multiple **Diaries**. (Usually a diary for each doctor at a practice)
- A **Diary** can have multiple **Bookings**. (Usually a booking for each consultation session a doctor has)
- **Bookings**:
 - A **Booking** can only be one **Booking.type**.
(e.g. *Consultation, Meeting, Follow-up*)
 - A **Booking** can be in any of the available **Booking.statuses** at a given time.
(e.g. *Booked, Arrived, Ready, Treated, Billed, Done*)
 - A **Booking** has both a **debtor** and a **patient**.
 - A **Booking** can only have one **debtor** and one **patient**.

- **Patient & Debtor:** (See also Figure 2 below)

- The **debtor** is the person responsible for payment.
- The **patient** is the person seeing the doctor. (Also known as the dependent of the debtor)
- The **debtor** and the **patient** can be the same person.
- One **debtor** can be responsible for multiple **patients** (or dependants).
- A typical example for the debtor/patient structure would be the following:
 - * A man has a wife and children. He is responsible for the medical account, and he is the main member of his family's medical aid. In this example the following is true:
 - **Debtor:** Man
 - **Patient(s):** Man, Wife and Children
 - * When the man goes to see the doctor, he is both the **debtor** and the **patient** of the booking.
 - * When his wife goes to see the doctor, he is still the **debtor**, however she is the **patient**.

See Figures 1 and 2 below for a visual representation of the various relationships explained above.

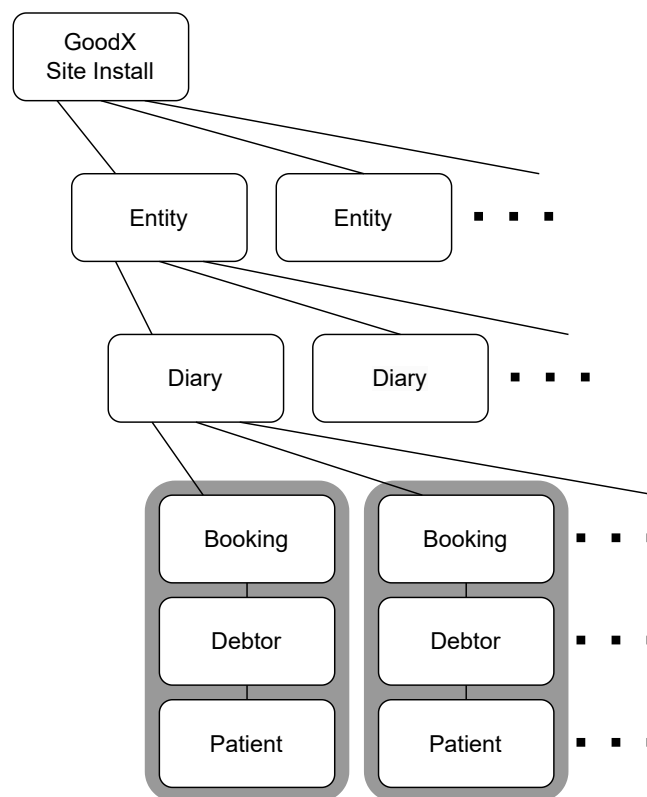


Figure 1: Entity, Diary, Booking relationship

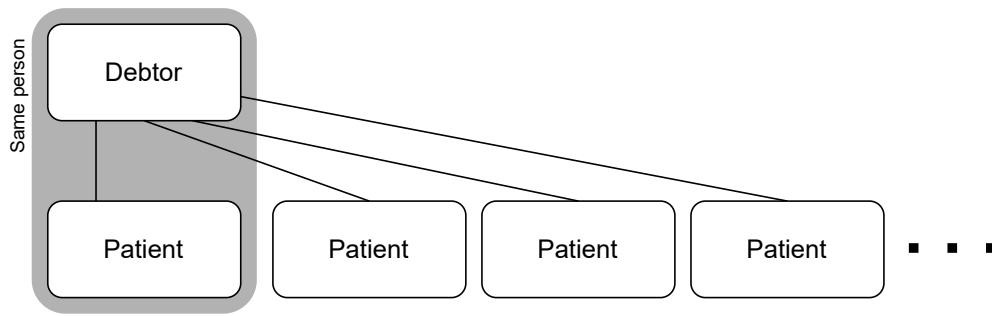


Figure 2: Debtor & Patient relationship

5.3 Requirements

The following minimal functionality has to be present in your application.
The ability to:

- Display bookings for the selected entity, for a given date.
 - There doesn't have to be an implemented date picker. You can choose to show bookings for the current day.
- Create a new booking.
 - Allow the user to select a patient from the existing patient list.
 - Allow the user to select the time and duration for the booking.
 - Allow the user to set a text reason for the booking.
- Update an existing booking.
 - Allow the user to set a different time and duration for the booking.
 - Allow the user to set a different reason for the booking.
- Delete a booking.
 - Prompt the user for confirmation.

5.4 Recommendations

- **Unit testing:** Consider adding unit tests to your application.
- **Code commenting:** Consider adding sufficient comments in your code for clarity.
- **Inspect Network activity in GXWeb:** Use the F12 DevTools in Google Chrome browser to monitor network requests and activity while exploring GXWeb. From there requests can be exported.
- **Validation:** Consider adding validation to data input fields where necessary.
- **Functionality:** While we value innovation, kindly bear in mind that our primary interest lies in observing your coding proficiency. Prioritise the fundamental functionality, and there is no need to allocate excessive effort to elaborate features or detailed data displays.

5.5 What's in the box? [1]

You should have received the following as part of your assignment:

- **GXWeb Site URL:** The URL for GXWeb which is also the base URL for the API.
- **GXWeb Login credentials:** A username and password to access GXWeb and the API.
- **Assignment PDF:** This document.
- **Postman collection JSON file:** Documentation for the API.
- (Optional) **Due date for assignment:** Please complete this assignment within 2 weeks, unless otherwise specified.

5.6 Resources

- **Postman Collection:** Use the provided Postman collection as documentation for the API.
- **GXWeb:** You can log in on our Web application using the same credentials needed to access the API. Here you are free to play around and explore.
- **GoodX YouTube channel:** You can visit this YouTube channel for tutorial videos for GoodXWeb. [5]

6 Submission

Please complete this assignment within 2 weeks, unless a different due date was provided by the sender of this document.

As part of your submission please return the following via email to the sender of this document:

- A link to a Git Repository containing:
 - Your application source files.
 - A "readme" file detailing running your application.
 - The Android Package file (apk), in the event you developed a mobile application.
- A link (and access details if necessary) to your website/app in the event you developed a web application.

Bibliography

- [1] A. K. Walker. (1995, September) Seven Movie, What's in the box? [Online]. Available: <https://www.youtube.com/watch?v=1giVzxyocIE>
- [2] GoodX Healthcare. (2023, November) About GoodX. [Online]. Available: <https://www.goodx.healthcare/about-us/>
- [3] A. Kumawat. (2023, April) What is an API (Application Programming Interface)? [Online]. Available: <https://www.geeksforgeeks.org/what-is-an-api/>
- [4] Codecademy. (2023, November) What is CRUD? [Online]. Available: <https://www.codecademy.com/article/what-is-crud>
- [5] GoodX Healthcare. (2023, November) GoodX YouTube Channel. [Online]. Available: <https://www.youtube.com/@GoodXSoftware>