

Functional Specification

bookDCU—Room Booking for DCU Staff & Students

Nacé van Wyk (20467546) & Scott Brady (20387871)

1. Introduction

1.1 Overview

1.2 Business Context

2. General Description

2.1 Product/System Functions

2.2 User Characteristics & Objectives

2.3 Operational Scenarios

2.4 Constraints

3. Functional Requirements

1. User Login

2. Booking a Room and Seat

3. Cancelling an Existing Booking

4. Modifying an Existing Booking

5. Verifying a Booking

4. System Architecture

5. High-Level Design

5.1 Sequence Diagram

5.2 Communication Diagram

5.3 Context Diagram

6. Preliminary Schedule

1. Introduction

1.1 Overview

The system to be developed shall be a booking system for computer labs and the laptop bar within the School of Computing. The system shall be called 'bookDCU'. The need for such a system arose after hearing feedback from other students as well as personal experience of how difficult it can be to find a space in a lab to work on project work, assignments or general study. Labs are often full and students are often left wondering around looking for a space to work, wasting valuable time.

1.2 Business Context

If proven to be useful and successful, the project could be acquired and used by the School of Computing in the future for reserving lab spaces.

The completed system could also potentially be of use to other universities or any stakeholders that would require a booking/reservation system, such as rent-a-desk co-working spaces, etc.

2. General Description

2.1 Product/System Functions

The system shall allow a DCU student to login with their DCU account, enter a date and time they wish to use a lab on and will return a list of available labs in which they can reserve a specific seat. This seat will then become unavailable for others to reserve for the duration of the current user's booking. The user must confirm their booking upon arrival in their chosen lab by means of scanning a QR code.

2.2 User Characteristics & Objectives

An average user of the system is any School of Computing student. As such, one would expect the users to have a reasonable level of knowledge of such a software system; however, the aim is to build a system that can be used by even those with the most limited of technical knowledge. This may enable the system to be ported and used by other faculties consisting of persons with less technical knowledge than those of the School of Computing.

The user simply wants a simple-to-use system that fulfils the purpose of enabling them to reserve a seat or lab at their convenience with minimal effort required on their part. In other words, we aim to implement a "design-for-all" approach when

designing and implementing the system, i.e. a user of the system will not require any special skills or training in order to make use of the system to its full potential.

2.3 Operational Scenarios

Main scenario: a student wishes to book a book a seat in a room on Wednesday November 23rd for 2 hours.

1. The student logs into the system and the DCU login service checks their credentials. If they are valid, they are let in and taken to the homepage.
2. The student selects 'New Booking' and enters the 23/11/2022 as the date and specifies the start and end time of the booking as 10am-12pm.
3. bookDCU returns the availability for that date and time and it is displayed to the student.
4. The student sees there is availability in lab L129 for the time he wants. He choses that room and selects seat 11.
5. His booking details are displayed for him so he can make sure everything is in order.
6. He submits his booking and it is confirmed.

Alternate scenario: a student wishes to cancel an existing booking.

1. The student logs into the system and the DCU login service checks their credentials. If they are valid, they are let in and taken to the homepage.
2. The student selects 'View Existing Bookings' and his bookings, if any, are displayed on his screen.
3. The student selects 'Cancel' next to the booking for 23/11/2022. The system confirms he wishes to cancel, and upon pressing submit, his booking is cancelled and released for another student to use.

- **Description:**

This will be the user's first interaction with bookDCU. The user will enter their credentials and click on the 'Login' button, which will bring them to the homepage if their credentials are valid.

- **Criticality:**

This is vital, as without logging in the user will be unable to access bookDCU and make use of its functionality.

- **Technical issues:**

As previously mentioned, getting ISS to co-operate with allowing us to use the "Login with your DCU account" system may prove impossible. As such, we may need to execute our contingency plan of using our own login system and use student emails/student numbers for login.

- **Dependencies with other requirements:**

N/A

2. Booking a Room and Seat

- **Description:**

After logging the user can choose the date and times they wish to book a room and an optional seat for. The system will return a list of rooms with space for that date and time, from which the user can choose one. Once they have chosen a room, they can choose a specific seat. They will then view the details of their booking to make sure everything is correct, then submit the booking.

- **Criticality:**

This is the primary function of the system, thus it is deemed essential.

- **Technical issues:**

Handling the date and time, i.e. making sure the date is not in the past, the start time is before the end time; displaying a clickable graphical seat map of each lab

- **Dependencies with other requirements:**

This requirement is dependent on Requirement 1, 'User Login', as a user must login before they can access the system and make a booking.

3. Cancelling an Existing Booking

- **Description:**

After logging the user can choose to manage their bookings if they have any.

The user will be presented with a list of their upcoming bookings. If they need to

cancel one for whatever reason, they can click the “Cancel” button beside it. This will release the booking and make the seat and time available for another student to book.

- **Criticality:**

This is an important aspect of the system to ensure that the availability of each room is maximised, though it is less important than being able to make a new booking.

- **Technical issues:**

Making sure the user cannot cancel a booking while it is active (i.e. cancelling a 2pm booking at 2:05pm on the same day).

- **Dependencies with other requirements:**

This requirement is dependent on Requirement 2, ‘Booking a Room and an Optional Seat’, as the details for the booking being cancelled need to be submitted in the exact same way as the details for a new booking would be set.

4. Modifying an Existing Booking

- **Description:**

After logging the user can choose to manage their bookings if they have any. The user will be presented with a list of their upcoming bookings. If they need to change the date and time of one for whatever reason, they can click the “Edit” button beside it. This will allow them to choose a new date and and/or time provided this new date/time is available for booking. If their chosen seat is still available, it will not change for the new booking. If it is not, the user will be asked to choose a new seat from those that are available. Once the user has finished modifying their booking, the new date and time will become unavailable for others to book and the original will become available for booking.

- **Criticality:**

This is an important aspect of the system to ensure that the availability of each room is maximised, though it is less important than being able to make a new booking.

- **Technical issues:**

Making sure the user cannot modify a booking while it is active (i.e. changing a 2pm booking to 4pm at 2:05pm on the same day).

- **Dependencies with other requirements:**

This requirement is dependent on Requirement 2, ‘Booking a Room and an

Optional Seat', as the new booking details need to be submitted in the exact same way as the details for a new booking would be set.

5. Verifying a Booking

- **Description:**

Upon arriving in a lab with a booking, the user will be required to verify they are using the booking by SCANNING A QR CODE???? in the lab in question, which will let the system know that this user is making use of the booking they have made.

- **Criticality:**

This is an important aspect of the system to ensure that it is not being abused by people making a lot of bookings and not using them, thus preventing others from making bookings.

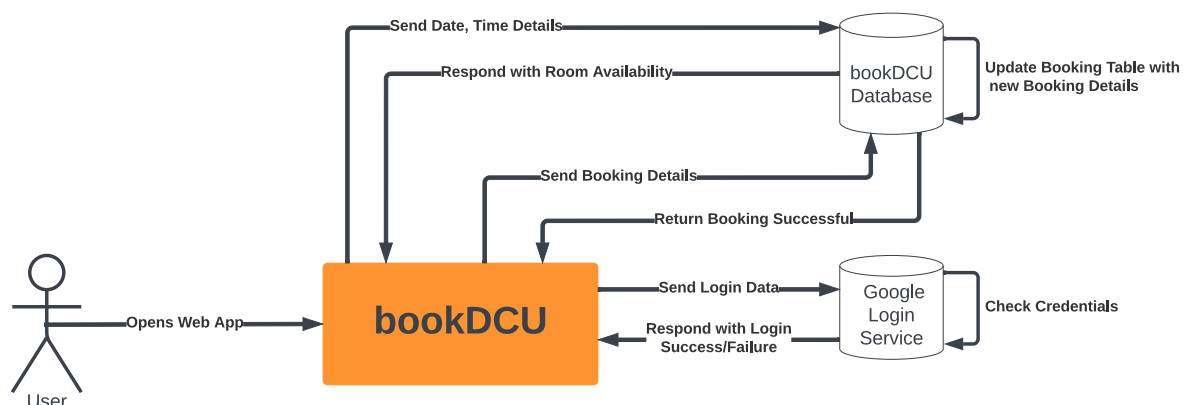
- **Technical issues:**

User will require a phone with camera in order to scan the QR code; being able to scan a QR code from a web app (possible using third-party software)

- **Dependencies with other requirements:**

This requirement is dependent on Requirement 2, 'Booking a Room and an Optional Seat', as a booking needs to exist in order for it to be verified.

4. System Architecture



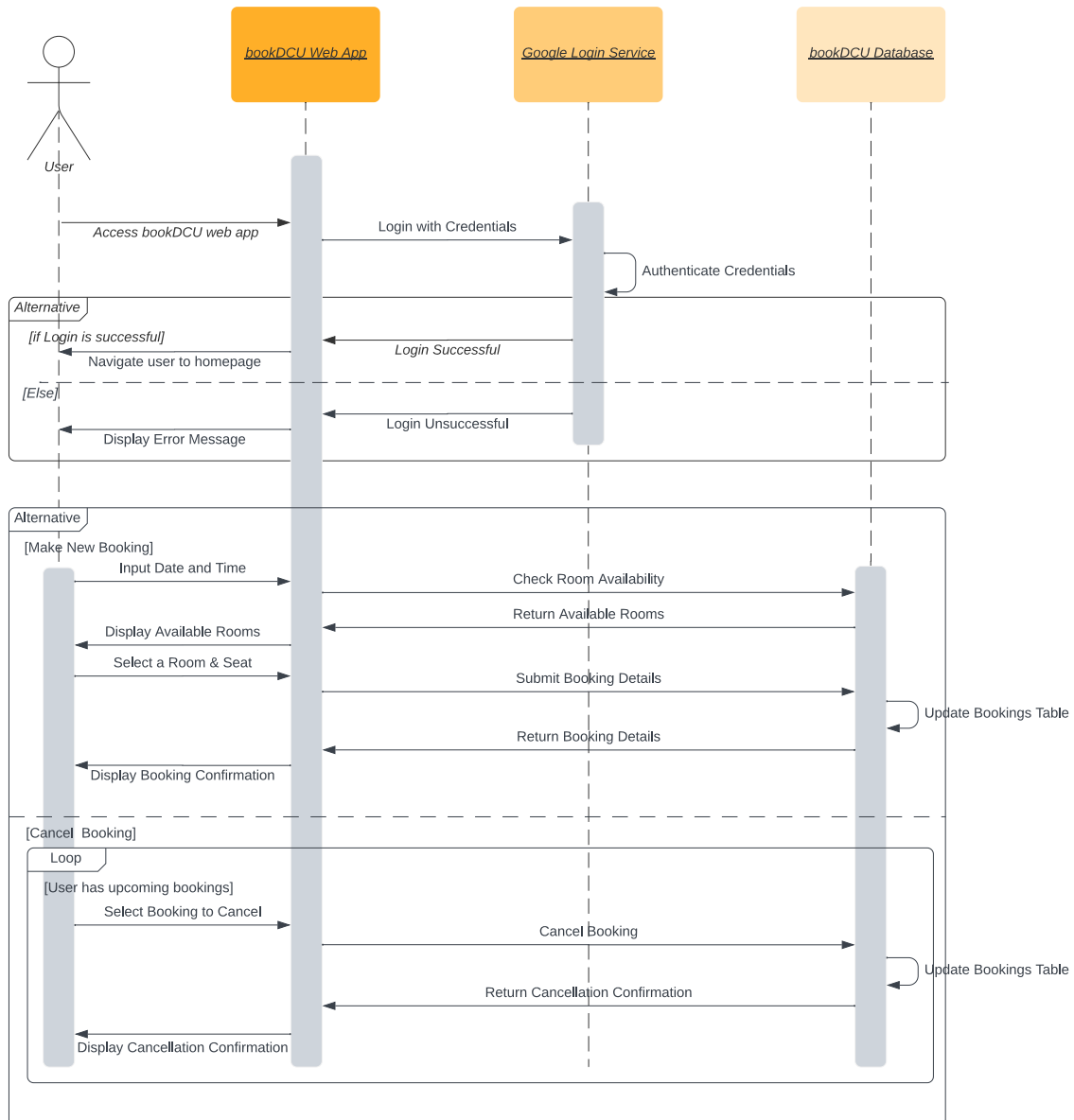
The system architecture is composed of three primary components: the bookDCU web application, the Google Login Service and the bookDCU database. These components are interdependent, with two components continually interacting with each other.

The user will open the bookDCU web application, which will then send the user's login credentials to the Google Login Service. The Google Login Service will verify the details are correct and inform the bookDCU web application.

The user can then choose a date and time for their booking which the bookDCU web application will send to the bookDCU database. The bookDCU database which will check the availability for the specified date and time and return available rooms to the user. When the user chooses a room and seat, the booking details are sent from the bookDCU web application to the bookDCU database and the bookings table is updated. Finally, the user is informed that their booking is successful via the bookDCU web application.

5. High-Level Design

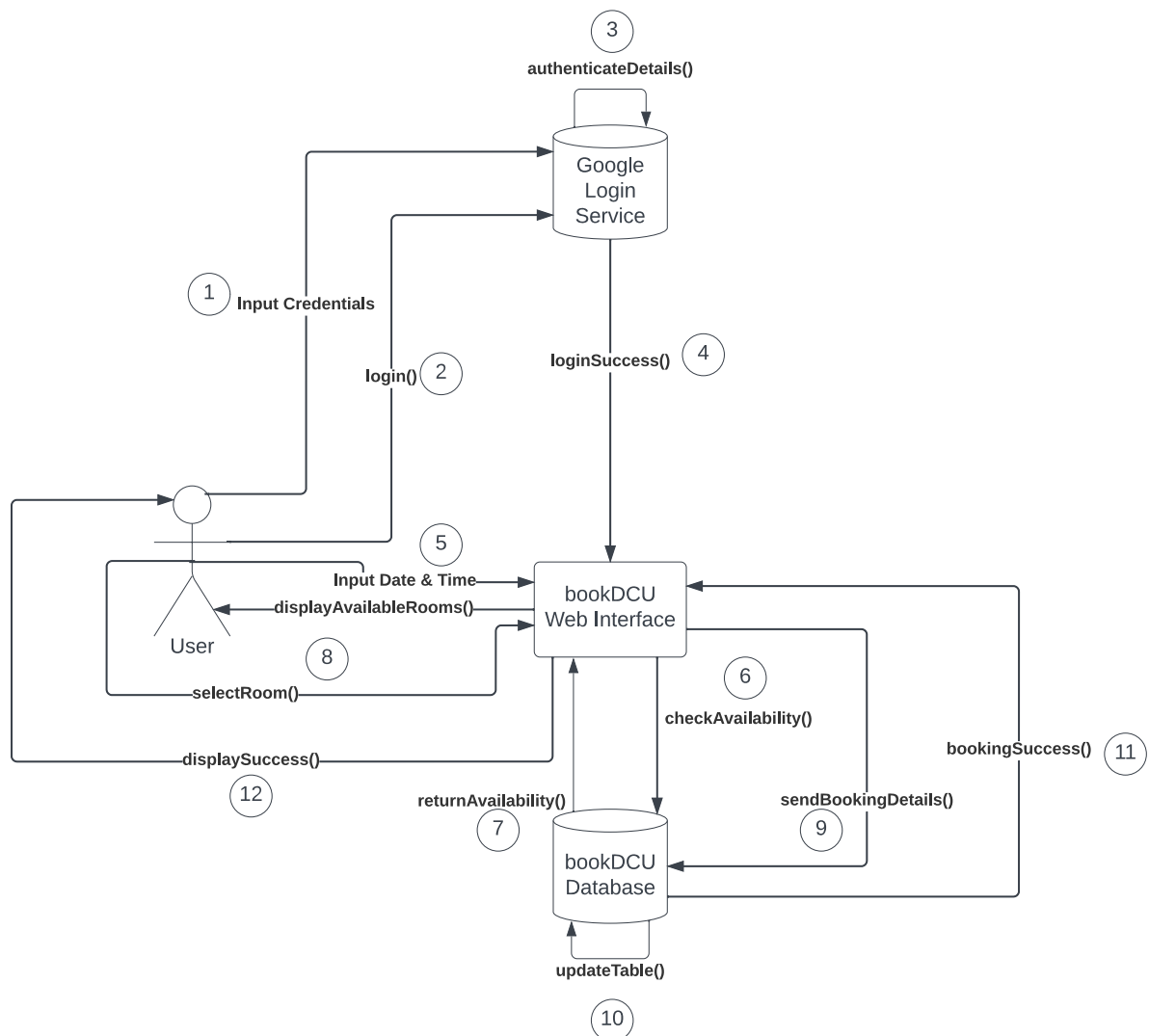
5.1 Sequence Diagram



In the above sequence diagram, the user logs in to bookDCU and if successful is taken to the homepage. They can then make a new booking by inputting a date and time. The bookDCU system will then check the availability and show what rooms are available to the user. The user can then choose a room and submit the booking. Following this the bookings table in the bookDCU database will be updated to reflect the new booking.

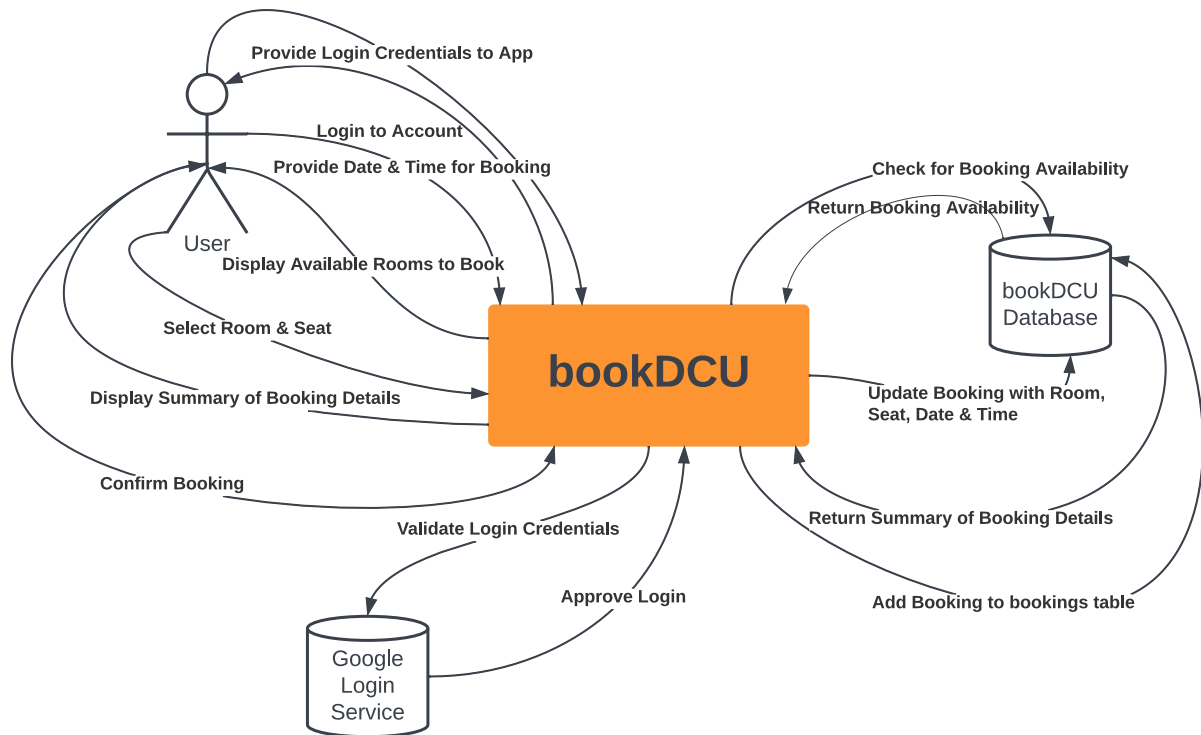
Cancelling a booking works in a similar fashion, except we must first check to make sure the user has any upcoming bookings before we allow them to cancel. Once the user chooses to cancel a booking, the bookings table is updated and confirmation of the cancellation is shown to the user.

5.2 Communication Diagram



The communication illustrates on a high level how the user, Google Login Service, web app and bookDCU database interact during the the process of logging in to the system, viewing rooms available to book, making a booking and receiving confirmation of the booking.

5.3 Context Diagram



In the bookDCU system, everything revolves around the bookDCU web app, shown in orange in the centre of the diagram. It illustrates how everything from logging in, getting room availability, making a booking and getting confirmation of a booking all flow through the web app. The web app makes use of the Google Login Service and the bookDCU database to make this all happen.

6. Preliminary Schedule

The Gantt chart below illustrates important milestones concerning the delivery of the final product and our proposed completion date for each milestone.

