Requirements Document
Club Resource Booking System

Student Organization Committee

Feb 5th, 2023

Contents

Revision History

Name	Date	Reason for Changes	Version
All	27.01.2023	Skeleton Document	0.0
Aleah	29`.01.2023	Section 1 Draft	0.1
Haley	30.01.2023	Section 2 Draft	0.2
Daniel	01.02.2023	Section 3 Draft	0.3
		Section 4 Draft	0.4
Kali Erickson	02.02.2023	Section 5 Draft	0.5
Haley Rutten	02.02.2023	Section 2 Draft Completion	0.6
All	03.02.2023	Final Edit	1.0
Kali Erickson	27.02.2023	Implement Change Requests	1.1

1 Introduction

1.1 Purpose

Currently, the University has an outdated and inconsistent system of booking rooms and resources for the purpose of club use at the University of Victoria. This document will propose a software solution to this problem. This software system will be a web application that can be utilized by club leaders at the University of Victoria for booking rooms and resources. The web app will interface with the current method of authentication used at the University of Victoria.

1.2 Project Scope

The new booking system will be a web application that will authenticate users via Netlink ID. The purpose of this application is to manage resources and room booking for club purposes. This would benefit clubs and their participants, as well as University staff overseeing these resources. Creating a central software solution would manage resources centrally, greatly reducing confusion. The goal of this solution is to make a platform that is user-friendly, robust and reliable.

1.3 Glossary of Terms

Term	**Definition**
Netlink ID	The credentials that identify staff and students at the University of Victoria
UVSS	University of Victoria Student Society

1.4 References

[1] "Netlink ID." *University of Victoria*, https://www.uvic.ca/systems/services/loginspasswords/netlinkid/.

1.5 Overview

This document will begin with the overall description of the project. This section includes the project perspective, project features, user classes and characteristics, operating environment, design implementation and constraints and assumptions and dependencies. Following that is the system features section. In this section, the various system features will be discussed. The next section will discuss the external interface requirements. This will touch on the user, hardware, software and communication interfaces our project will interact with. The next section will overview the other functional requirements such as performance, safety, security and software quality requirements. Finally, the document will discuss the other requirements necessary for this project.

2 Overall Description

2.1 Product Perspective

Currently, the University of Victoria lacks a streamlined, responsive booking system for club resources. The perspective for the solution is as follows:

- The product will replace the current booking form accessed through UVSS.
- The product will shift from a form to an automated booking system, providing the user with a quick response.
- The application will keep track of the inventory of available resources that the University provides. The inventory will have to be modified by an administrative member of the university when new resources are purchased or old resources become unusable.

2.2 Product Features

The club booking system will allow registered club members to book meeting spaces and equipment (such as tables, chairs, projectors, and speakers). The system will rely on the University of Victoria sign-on services to validate users. It will also allow users to export meeting times to their google calendar, and keep a list of a user's current bookings.

2.3 User Classes and Characteristics

There will be two classes of users for this application, student and admin. The main focus of the project will be the student side of the application, as they are expected to generally have less technical expertise and will have a larger user base.

2.3.1 Student Users

The main users of the system will be students of the University of Victoria. These students must be members of clubs registered with the university to use the system. They will be allowed to view bookings related to their club and book available resources. Students will also be allowed to cancel bookings for their club. The two different roles and associated permissions for student users are "Club Leader" and "Club Member". The Club Leaders will have permissions to add and cancel bookings on behalf of their club, while the Club Members will only have permissions to view the available and current bookings.

2.3.2 Admin Users

The second type of users will be administrators from the University of Victoria. Admin users will have elevated permissions, and be allowed to view and cancel all bookings across clubs in case of errors or abuse of the booking system. They will also be allowed to add and delete items from the inventory, as the resources available will change over time.

2.4 Operating Environment

- Hardware Platform: The application will run on laptops and desktops, but will not be optimized for use via smartphone due to time constraints.
- OS and versions: The application should run on any internet browser.It should work on any operating system.

2.5 Design and Implementation Constraints

Maintenance of Software
The system will be updated and maintained by the SOC development team. If unauthorized changes are made to the software by a third party or the client, SOC will be released of maintenance responsibilities.

Database
The system will rely on a database of resources provided and maintained by the University of Victoria. The system will be developed using mock data.

Timing Requirements
Since more than one user may try to book the same resource simultaneously, race conditions will have to be considered in the development process.

2.6 Assumptions and dependencies

Internet Connection
The application will depend on an internet connection

Admin Maintenance of Database
The system's effectiveness will depend on being an accurate inventory, which will have to be maintained by Admin staff from the client.

Availability of Client

Since the system will be relying on a database provided by UVic, open contact must be provided to ensure a compatible system is developed.

3 System Features

3.1 User Account

3.1.1 Description and Priority

In order to use our platform's services users need to be registered. This is accomplished by requiring every user to access the application's services through their own private account. The user's account should be linked to their student ID and respective club. Users should be able to manage their bookings through their accounts. While this feature is important from an administration perspective it is not needed for the core application to function and is thus of a **medium priority**.

3.1.2 Functional Requirements

Req-1-1: User's accounts need to be accessed through their Netlink ID.

Req-1-2: Accounts need to be able to show which club(s) the user is representing.

Req-1-3: Users should be able to sign up for membership in multiple clubs.

Req-1-4: Each account should have an account page with the user's info, including club position(s), and bookings.

Req-1-5: Users should be able to see all the bookings they made in their account on their account page and which club the booking was made for.

Req-1-6: A club should not have more than one account representing them. The active account connected to a specific club must be deactivated before a new account connected to that club can be made.

Req-1-7: Users with Club Leader permissions should be able to modify their bookings from their account page

3.2 Item/Room Search

3.2.1 Description and Priority

Our application should have a feature allowing users to look up all items and rooms UVIC provides for student clubs. When a user searches for an item/room all objects fitting the search category should be displayed. This search feature is at the core of our application making it a **high priority** as the application won't work without it

3.2.2 Functional Requirements

Req-2-1: Searches should have the following keywords: Name, Type, Booking Time Period, Location

Req-2-2: Search results should only show objects available during specified Booking Time Period

Req-2-3: Search results should be sorted in order by Key Words Accuracy, Popularity then Alphabetically.

Req-2-4: Search results should appear in the “booking fields”. These boxes include info about the specific item and are specified in Item Booking Fields and Room Booking Fields.

3.3 Item Booking Fields

3.3.1 Description and Priority

As our application revolves around student clubs booking material from UVIC, it is a high priority to ensure this feature is fast and easy to use. In combination with the search feature users should look up items according to the given search categories and then book the needed amount while specifying the use.

3.3.2 Functional Requirements

Req-1-1: Items will be booked from the results page, that appears after completing a search query using the booking fields.

Req-1-2: The User has to input the amount of that item they require in order to complete the booking

Req-1-3: The User is required to input the purpose for the booking

Req-1-4: There should be a check box the user has to press confirming they have understood the rules required to book the given item.

Req-1-5: There should be a “Book” button the user needs to press to complete the booking

3.4 Room Booking Fields

3.4.1 Description and Priority

This feature works similarly to Item Booking and comes attached with the same **high priority** requirement as this feature is needed for our application to fulfill its purpose. However, since each room is unique and can not be displaced it comes with different functional requirements than the Item booking feature.

3.4.2 Functional Requirements

Req-4-1: Rooms will be booked from the results page, which appears after completing a search query using the booking fields.

Req-4-2: The User is required to input the purpose for the booking

Req-4-3: There should be a check box the user has to press confirming they have understood the rules required to book the given room.

Req-4-4: There should be a “Book” button the user need to press to complete the booking

3.5 Admin Access

3.5.1 Description and Priority

This feature should allow the administration to view and modify all bookings. This feature is needed so that it is easy to fix any mistakes that might occur during the booking process or if any special arrangements need to be made. While the application can function without this feature, its functionality is still important for the long-term health of the program. Thus it has a **medium priority**.

3.5.2 Functional Requirements

Req-5-1: Admins should be able to create an Administrator account

Req-5-2: Users with an Admin account should be able to change the properties of any items in the database

Req-5-3: Users with an Admin account should be able to view all other account pages

Req-5-4: Users with an Admin account should be able to modify the bookings of any other account

Req-5-5: Users with an Admin account should be able to delete any other account

Req-5-6: Admin account should be able to view booking statistics

3.6 System Memory

3.6.1 Description and Priority

The system should be able to memorize which items have been booked through the lifespan of the application with the purpose of being able to recommend popular items to users when searching for objects to book. As this would increase the general speed at which users are able to find what they are looking for. Additionally, it would allow UVIC to determine which items they might want to buy more off and which are less needed. This should all be done without compromising the data privacy of individual users. This feature has a **low priority** as it's not needed for the application to function.

3.6.2 Functional Requirements

Req-6-1: Record the number of times an item or room gets booked.

Req-6-2: Record if a specific type of item ever gets booked out.

Req-6-3: Record the length that items and rooms are booked for.

Req-6-4: Communicate this information to the search function in real time

4 External Interface Requirements

4.1 User Interfaces

The user interfaces in the club booking system will be inherently similar to existing scheduling systems. Some examples of such user interfaces include, but are not limited to, the following:

- One interface tailored for website use
- A page to query for available rooms, and equipment
- An interface to add or remove previously made bookings.

4.2 Hardware Interfaces

The system will be implemented as a web application. The web application will support all major current web browsers across all modern operating systems. Navigation will be supported with a mouse and keyboard on the web browser and via touch navigation on mobile devices.

4.3 Software Interfaces

A club administrator visits our page and logs in via their Netlink ID. These requests interact between UVic and the club administrator's web browser. Once authentication is confirmed, the user can now reserve rooms, and equipment on behalf of their club. The web interface will be designed using a to be determined web framework. The mobile application will interact with the base mobile operating system directly.

The internal system will require interaction with a yet to be determined database software for storing room information and equipment. The product will also likely rely on UVic for hosting the web interface.

4.4 Communications Interfaces

The club booking system will require integration with UVic to authenticate users. This will be done by transmitting data over HTTPS to uvic's internal systems using industry-standard internet encryption standards such as SSL and TLS.

5 Other Non-Functional Requirements

This section will describe the non-functional requirements of the “Club Resource Booking System” project proposed by the Student Organization Committee development team.

5.1 Performance Requirements

The performance requirements for this system are concerned with ensuring high-speed performance. The students and university administrators who will be using the system have busy schedules and need a solution that will decrease the amount of time it takes to book and manage club resources. Therefore, the system must have a response time of less than 0.1 seconds, so that results feel instantaneous to the user. This means that the system should be designed to have a lightweight UI to reduce bandwidth consumption and should avoid using redirects. Additionally, the software must be available for 12 hours a day and so must be tested for over 100 hours a week.

5.2 Security Requirements

The main potential security risks arising from the integration of the system are privacy related. In order to ensure that only students with the right permissions afforded by their position in their club can book resources, the system must store information related to each student's current club involvement. To minimize risks, the system will only store Netlink IDs paired with club positions. No other student information (student IDs, passwords, full names) will be stored by the databases necessary for this system. When this information is stored, it must be encrypted to ensure privacy in the case of wrongful access. Additionally, only university administrators will have access to the stored information and must use the Netlink-ID with existing university databases to access other data. In order to ensure this requirement is met, thorough access control testing must be conducted.

5.3 Software Quality Attributes

Development must prioritize scalability and usability. Scalability with reference to the amount and types of data is paramount to ensure that new clubs and corresponding positions, or new or greater volumes of resources will be easy to enter into the system. Ease of use is the most important usability factor for this system because students who participate in extracurricular activities have busy schedules and need the system to make the resource booking process simple and fast. Users who have experience with the system must be able to complete the booking of a resource within three minutes. Furthermore, the accuracy of the self-adapting resource suggestion lists based on the frequency of bookings must be tested to ensure that the feature has the desired effect of increasing ease of use.

Appendix: Issues List

Integration with the UVIC database and login in system has yet to be determined.

1. Separate Students into two subtypes, a "Club Member" and a "Club Leader"
    a. A club member is able to view bookings, and a leader is able to book them
2. Have a way for a student to join club(s)