Requirement Specification

Team Pulse on behalf of RGU enterprise & Innovation Hub

Enterprise Space Booking System

Table of Contents

[Abstract 1](#_Toc84557438)

[Team Composition 2](#_Toc84557439)

[Requirements Categorisation 3](#_Toc84557445)

[Functional Requirements 3](#_Toc84557446)

[Non-functional Requirements 5](#_Toc84557451)

[Product Backlog 6](#_Toc84557456)

[Testing 6](#_Toc84557457)

[References 7](#_Toc84557458)

# Abstract

This document details the initial preparation of a project being undertaken by Team Pulse, on behalf of our client Robert Gordon University (RGU) Enterprise and Innovation Hub.

The goal of Team Pulse is to centralize and simplify our client's booking system for their working spaces/rooms available at their on-campus innovation hub, and city centre enterprise space respectively.

We at Team Pulse, have decided to approach this from the direction of maintaining simplicity to the end user, with the goal of streamlining the process of booking and related administerial needs not only for the end user, but also for operators too.

Our desired result is to improve the experience for administrators, where they can more effectively utilise these enterprise spaces, while also enabling easier streamlined access to the booking process, rather than the pre-existing solution which places more of the work of operations on administration staff.

This project shall be implemented using agile software development scrum methodology, utilising a product backlog for objective tracking over three sprints. This should enable our team to rapidly develop the product and improve efficiency in the implementation process.

# Team Composition

Team Pulse is comprised of 6 individuals, from a variety of differing sectors within RGU’s School of computing, with a diverse range of backgrounds and experience, this will be utilised by the team to our advantage, as we will focus each team members efforts on the tasks, they enjoy that they feel most experienced in across our range of development, to effectively play to our strengths as much as possible, with other team members assisting as and when required/when time constraints allow.

Listed below are the team members, with their respective course/discipline and Belbin Role test (123test, 2021) results, to give an overview on character, with a brief description of each members contributions.

Jon A. Beroz - Computing Science, Analyst/Expert

Enriches the team with dynamic information around emerging frameworks, highlighting those which may maximise outputs from the team. Coming from the experience gained during their web focused internship, sharing deep insight on web development and technologies that can be used for each task, providing the team with the optimum frameworks where appropriate.

Matthew Brougham - Digital Media, Driver

Elder of Team Pulse, unleashing a stream of possibilities to consider in future for the team, while contributing to Design and Interface implementations. Editor for the team's documentation, while offering industry experience in client handling and real development projects.

Percy Lackgren - Computing Science, Analyst/Chairperson

Challenges the ideas brought forward by the group with the goal of achieving watertight solutions, while also providing experience from a year work placement in web development, database design, client interactions, system documentation, and real-world project experience.

Laura Mckell - Cyber Security, Innovator/Explorer

Passionate about cybersecurity and development, equipping the team with an extensive knowledge base in security relating to web and databases, bringing problem-solving skills and out the box thinking while striving to maintain a harmonious environment within the team.

### Oli Prajwolshree - Digital Media, Driver/Innovator

Furnishes the team with a multitude of diverse propositions, troubleshooting and researching into the further development and visual layouts of project tasks. Enriching the teams experience with design and photography combining both into modern experiences for end users of the team's efforts.

Allegra Santoro - Computer Science, Analyst/Driver

Aspiring developer with exceptional organisational and observational skills. Adept at both back-end and front-end development, pushing for excellence across both. Tasked with the role of scrum master, ensuring accurate and comprehensive records are taken for the duration of the project.

# Requirements Categorisation

The requirements of this project are categorized according to their functionality as either, Functional or Non-Functional, these two functional groups are then prioritised according to MoSCoW.

Functional requirements define the base requirements that must be satisfied for the system to function as intended while Non-functional requirements define the non-essential requirements that specify how the system can achieve its functional component (QRA Corp, 2019).

MoSCoW is prioritisation method which separates requirements into, Must-have, Should-have, Could-have, and Won’t-have (Product plan, 2018).

Must-have requirements are required for the system to function as an MVP (minimum viable project), if any of these are not integrated the system will not function as intended.

Should-have requirements while not vital for the functions of the system and are not vital for a MVP, however they could bring great value to the system if implemented.

Could-have requirements are like Should-have requirements whereas they are not required for a MVP, they also bring less value when implemented can be the first to be deprioritised if necessary.

Won’t-have requirements are features that will not be a focus for the current system as they are out of scope or infeasible for the current timeframe or project plan.

## Functional Requirements

### Must-Have

* The system must have a home page.
* The system must allow the user to select the location of the booking.
* The system must allow the users to select a date and time range for the booking.
* When a new time and date range is entered, the room availability display must update automatically.
* The user must be able to complete a booking.
* A logged in user must be able to cancel their bookings.
* The system must require the following fields in order to create a booking:
  + Full name
  + Email address
  + Phone Number
  + Role at the university (if applicable)
  + Purpose of booking
* The system must allow users to create an account.
* Creating an account must request the following:
  + Full name
  + Email address
  + Role at the university (if applicable)
  + A password
* The system must allow users to login/logout with their account.
* The login must request the following from the user:
  + The email address connected to their account.
  + The password of their account.
* The system must allow users to delete their account.
* A user’s data must be removed from database once their account is deleted.
* The system must require confirmation for whole room bookings by administrators.
* The system must send booking validation emails to users.
* The system must display all room information in real time.
  + Calendar.
  + Room availability.
* The system must allow the user to select between a full room and shared room booking if the room allows it.
* The system must not allow bookings that exceed the room’s currently available slots.
* After completing a booking, the website must confirm to the user that the booking request has been completed successfully.
* If the user requested a full room, the page must explain that a member of staff will review their request.
* The website must display an Outlook calendar
  + The calendar must be filtered by room.
  + The calendar must display the bookings in the selected room.
  + This calendar must only be accessible to administrators.
* Once a booking has been confirmed it must be added to the calendar of the chosen room.
* A logged in user must be able to access a page containing all their bookings.
* An administrator must be able to see all the bookings needing approval.
* An administrator must be able to approve or reject bookings.
* An administrator must be able to create bookings without approval.
* An administrator must be able to edit or delete existing bookings.
* The administrators must be able to access a page with all the bookings within a date-time range and location.
* There must be a database to store the required data
* Passwords must be hashed before being stored in the database.
* The system must encrypt the connection when a password is transmitted over a network.
* The system must logout a user after an hour of inactivity.

### Should-Have

* The system should display room bookings statistics to administrators.
* The user should be able to edit their bookings.
  + If it is a full room booking, the edit must be approved by an administrator.
* The user should be able to reset their password.
* The system should allow users to edit their account.
* If an administrator rejects a booking, they should be able to specify why.
* An administrator should be able to edit and create rooms.
* The system should not redirect to any third-party websites.
* The database should use a cryptographic has such as SHA-512.

### Could-Have

* The system could allow bookings to be set to be reoccurring.
* The system could allow for further additional services for the room to be booked (e.g., A display)
* The system could send text alerts to users with a booking on the day of their booking.
* The system could display images of the rooms.
* The system could allow users to book specific seats/stations.
* The system could allow users to book for others.
* The user could be able to opt in for a “remember me” option for their username and password.
* The system could utilise SSL/TLS connections.
* The system could utilise 2 factor authentication.

### Won’t-Have

* The system won’t use artificial intelligence to suggest a booking.
* The system won’t provide NHS track and trace integration
* The database won’t use a “salt” of an 8-character random sequence.

## Non-functional Requirements

### Must-Have

* The booking system must be a website
* The website system must be accessible from mobile
* The website must adapt to mobile view.
* The website system must be accessible from a desktop device
* The website must be compatible with iOS.
* The website must be compatible with Windows devices.
* Data must be handled in accordance with GDPR.
* The system must only accept passwords between 8 and 21 characters.
* The system must only accept passwords with at least one use of special characters and one use of a number.
* The system must hide a user’s password when they are entering it.
* The system administrator must be able to reset passwords.
* Admin users must be able to see the full name, email, and phone number of anyone who has made a booking.
* The system must be checked frequently for security vulnerabilities.
* Users of the system must have a guidance document.
* Non-admin users must not be able to see information from other users.

### Should-Have

* The system should use safe-for-web colour.
* The user should validate their email before using their account.
* The “reset password” function should send a password reset email to the user’s email account.
* A user should only have one session open simultaneously.
* All password fields should be marked as “passwords” in HTML.

### Could-Have

* The system could utilise captchas at login.
* The system could lock out a user if their authentication is incorrect 5 times in a row.
* The system could support users who have cookies disabled.
* The system could have a failure secure if the database or web service is down.

### Won’t-Have

* The system won’t be translated to other languages.
* The system won’t be invulnerable to rapid-fire login attempts.
* The system won't be invulnerable to SQL injection attacks.
* The system won't have a vulnerability assessment.

# Product Backlog

A product backlog is an essential step to “bridge the gap between the processes of generating user stories and realizing them in working code” (T. Sedano, 2019). It serves as an indispensable tool to maximise task prioritisation and management by arranging them in an ordered list.

The current backlog of this project contains a brief description of each task accompanied by its task ID, current status, expected duration, team members responsible and the tasks individual prioritisation. The latter is a temporary indicative estimate, which will need to be discussed and confirmed by the client. The list also assigns a level of dependency to each task, the lower the level, the fewer tasks the current one depends on to be completed. For instance, if a task depends on the completion of a level 2 task to be finalised, this will be assigned a dependency level of at least 3. The prerequisite tasks are also listed in the product backlog. This system will provide the team with an additional tool to finalise tasks prioritisation.

(Product backlog available in document “A1\_Pulse\_Product\_Backlog”)

# Testing

In the development of this project, we will utilise a variety of testing methodologies throughout the development process.

Manual unit testing will be utilised throughout the initial development process to ensure each component operates correctly, in combination with white box testing to strengthen the results of the testing.

Continuing into development when these components are combined, Integration testing will take place, to confirm the compatibility of various features and components in tandem, to maintain system performance.

In the final stages of the implementation process, system testing will take place, utilising black box testing to ensure the end user experience is as smooth as possible.

As the implementation phase comes to an end we shall perform performance testing, to evaluate the system is fit for purpose and operates as it should, the aim is to achieve this via a variety of tests, including but not limited to load testing and spike testing,

Following this, we shall aim to test further in security and usability, taking GDPR and ease of use and ergonomics into consideration.

# References

123test, 2021. *Team roles test | take this free team roles test online at 123test.com.* [Online]   
Available at: https://www.123test.com/team-roles-test/

Product plan, 2018. *What is MoSCoW Prioritization? | Overview of the MoSCoW Method.* [Online]   
Available at: https://www.productplan.com/glossary/moscow-prioritization/

QRA Corp, 2019. *Functional vs Non-Functional Requirements: The Definitive Guide - QRA Corp.* [Online]   
Available at: https://qracorp.com/functional-vs-non-functional-requirements/

T. Sedano, P. R. a. C. P., 2019. The product backlog. *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE),* pp. 200-211.