Requirement Specification

for

Park View Community Centre Website and Booking System

Prepared By

Team 6
Erdal Guclu
Rowan Ho
Nikesh Patel
Greg Stretton
Cameron Tibbles
Jack Tweddell

Version: 1.0 | Date: 16/11/18

Contents

Overview and Justification

Park View Community Centre (PVCC), in partnership with Park View School (PVS), is an organisation in Chester-le-Street that makes use of facilities and events spaces at their two locations. They take bookings for sporting activities along with the use of their theatre, teaching, cinema spaces and other bespoke requests.

The Problem

PVCC has a website which they feel is outdated. Currently, customers can see a calendar of upcoming community centre events, and view facility descriptions. However, to book facilities, they have to send their details in a web form to the centre. The booking is then entered manually by staff into the community centre's Google calendar. Customers also have to pay for their bookings in person when they arrive.

Furthermore, staff have to manually check the PVS schedule for potential clashes, as they share facilities with PVCC.

Proposed Solution

Our proposed system makes updates to the PVCC website. This includes:

- ➔ Implementing a booking system which allows customers to place a booking of a PVCC facility online.
- ➔ A payment system linked to the website which allows users to pay for bookings online, or over the phone.
- ➔ Combining the calendar currently on the site with the calendar of PVS.
- ➔ Making other design improvements to the site that would improve user experience (UX).

This will benefit not only the users of the PVCC website, such as residents of the town, through easier bookings and better UX, but also the admins at PVCC through a reduced workload.

Project Scope

System Overview

The project scope involves the calendar, booking system and the addition of an online payment (eg. PayPal) system for the PVCC website.

The overall design of the website provides a strong baseline to build upon. However, the client has stated that the website feels “outdated” so we have looked over the existing system and found certain areas of improvement.

The website will be capable of showing at a glance all the events at both PVS and PVCC, allowing easy booking of most activities in the facilities available. It will also allow for secure online transactions to occur in order to facilitate the bookings.

PVCC will provide us with a set of basic activities available in their facilities to allow for fully automated bookings. However we also expect customers to have more complex requirements for activities that may not have a set price or may require multiple facilities, which will require manual oversight by the PVCC staff.

The new improved website should significantly reduce the workload of PVCC staff by allowing most of the bookings to be automated and by unifying the Google Calendar of PVS and PVCC so that everyone can at a glance see availabilities.

Stakeholders

PVCC Admins

The admins will manage any bespoke requests that require human oversight and so they will be accessing the calendar - which is linked with PVS and they will also be required to do event promotions, which means that they will modify descriptions on their site.

Families and Community Members

This group will only be interacting with the booking and payment system, at least where user input is concerned. Other than that they are expected to navigate the site to check out events and facilities and be able to check (but not edit) the calendar.

PVS

All of the facilities of PVCC are owned by PVS and as such they are a priority client of PVCC. The school uses these facilities to host everything from events to exams and as such their calendar will be linked with that of PVCC.

System Description

Domain of the Project

The client has requested we make improvements to their current website, therefore the main domain of this project is that of web applications. A sub category this project fits into is that of booking systems. The type of booking system we aim to build is a 'rental services' system, unlike a 'hotel' or a 'ticket system' which feature non-specific bookings and no concept of service duration, respectively.

Case Studies

Case Study 1: Trainline Booking[1]

Trainline is a website that allows you to book train tickets, therefore it features a booking system and is relevant as a case study.

A prominent feature of this, and almost all online booking systems, is the use of a Real Time Transaction System (RTTS). The basic principle of the RTTS is that when a user begins booking a slot, the slot is locked out in the database until the transaction is completed or aborted. This ensures double booking does not occur, as single fields in a database are not allowed simultaneous access.

Another key feature of 'www.thetrainline.com' is the quality of the GUI design. The GUI has multiple features that could serve our implementation well:

- ➔ Text boxes featuring autocomplete to enhance ease of use.
- ➔ A calendar to easily select dates and times.
- ➔ Use of combination boxes and radial buttons to limit choice.

Trainline is a ticket booking service whereas our system will be a rental service. Therefore we need a concept of duration attached to each booking, unlike in a ticket service.

Features not included/not applicable:

- ➔ Need a 'time' calendar - slots within days
- ➔ Scale is months - our scale is days
- ➔ Time slots are important

Getting an exact date and time from a searching user simplifies the system a lot, but it limits ease of use. For example, if the user just wants to have a look at what's available and doesn't know specifics yet, prompting for a precise time and date can be off putting. Trainline shows availability either side of the specified time to give more flexibility, but if the user could see all availability for a large time period it would be even better. By implementing an interactive calendar, the website can show the user all availability for a given week so that they can choose what is best for them.

Case Study 2: Bill Bryson Library Study Rooms Booking[2]

The 'Bill Bryson Library' is the university library for Durham. One of the facilities it provides is the use of its study rooms. As these rooms are in high demand, their use is allocated via a booking system based on the library's site.

The website shows availability of individual study rooms, but this data doesn't need to be exposed to the user. Study rooms that are identical could be grouped together, and the availability information combined. Specific details such as "music listening" could be specified with a checkbox. This way, the user could specify just a few options and see available times, instead of having to check 21 separate rooms. We can apply this principle to identical facilities if applicable, for example classroom bookings.

The layout of the website leaves something to be desired. The design is a simple, plain bullet point list with hyperlinks to each specific rooms booking page.

If this data were condensed into a single calendar, it would have the benefits of providing a single, visual representation to make it easier to use, with the added bonus of being aesthetically pleasing

Case Study 3: HSS Hire[3]

HSS is a website that rents out trade tools and equipment to users. This site is of interest to us because it has both automated and manual booking features. For example, small items such as portable air conditioners can be booked automatically using a traditional basket and checkout system. However, for more significant items like large cooling systems, the company has to be directly contacted e.g. telephone, or an enquiry made through an online form.

The enquiry form begins with an explanation of how to make an enquiry, and lists contact details as the first method of making an enquiry. We can incorporate this into our design by displaying/linking PVCC contact details on bespoke booking requests. The second method of enquiry is via a form located directly underneath the explanation paragraph. The form is simple, containing basic text fields to enter information, with an easy to use calendar to select dates. We can use the calendar feature to allow users to select relevant dates. However their form requires a lot of information that would be redundant for our purposes, so we would only collect necessary information.

Overview of system to be built

Booking System

The current system is enquiry based, with a form to fill in that gets emailed to PVCC. We will keep this functionality for advanced bookings that can't easily be automated. For example, events that require multiple rooms set up in specific ways would be easier to organise through an enquiry system; this feature has been requested personally by the client. The enquiry will be through a simple form with a name, email, phone number, and message section that will be sent to PVCC through email for a PVCC admin to do a manual booking.

Our proposed system would add additional functionality to this process in the form of an automated booking system. This would allow users to book PVCC facilities with minimal/no interaction needed from PVCC staff members. Booking systems can introduce potential problems, with one of the most likely being double bookings. In order circumvent this, our booking system will be implemented as a RTTS, ensuring double bookings cannot be made. Furthermore, basic bookings will now have the option to pay online via PayPal. The user will be able to enter their details into PayPal's payment system, which will be processed and then deposited into PVCC's account (minus a transaction fee). Security is always a concern when handling card payments, so by using a verified middleman, we do not have to handle or store card details - which is safer for the user and has less liability for PVCC.

Calendar

The site currently features a basic calendar that shows what events are on.

Behind the scenes, the system currently involves PVCC staff manually inputting booking data into a calendar on the site. However the school has a Google Calendar, and there can be conflicts since both organisations can book the same facilities. We will synchronise the calendars of the community centre and the school to remove the risk of double booking and reduce the time wasted by PVCC staff having to manually sync the calendars. We will do this by merging the calendars.

Redesign of Site

The aesthetics of the site are modern and friendly. However, upon further inspection, the site has navigational flaws and has potential for aesthetic improvement. On the navigational side, key pieces of information are hard to find and take too many clicks to reach e.g. in order to see the facility price list, you must go through the 'What's On' and 'Planning something?' submenus when it should be located on the nav bar or under facilities. On the visual side, the home page is cluttered with things that would be better placed elsewhere, such as the 'Easy Fundraising' section being under its own tab. There are also other improvements that could be made - Twitter integration for example. In short our system will include multiple changes that while preserving the original feel of the site, will vastly improve the user experience.

Website Protocol

The website is currently using HTTP to handle user requests. This poses a problem when sending data such as form submission and especially payment details. For this reason, we will upgrade the websites security to HTTPS.

Solution Requirements

Functional requirements

Our functional requirements can be split into 4 parts:

FR1 - The booking and payment system

FR2 - Synchronising the google calendar

FR3 - The redesign of the website

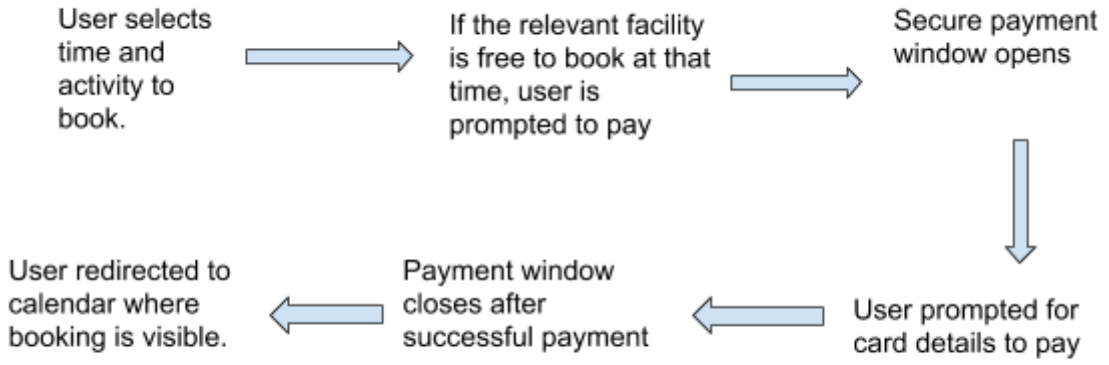
FR4 - HTTP to HTTPS

Where necessary, individual functional requirements have been broken down into subsections.

ID - Title	FR1.1 - Online Automated Booking System
Description	Customers should be able to book certain Park View facilities online when they are available. Selection of the booking, payment and confirmation should all be fully automated.
Dependencies	FR1.2, FR2, FR4
Priority	High
Expected Results	The user should be able to select the facility they want, specify what they want it for, then place a booking request online. If this booking request is at a time when the facility is freely available the customer should be prompted by the website to confirm the booking, and be asked to pay via the online payment system. (FR1.2) After successful payment, the user's booking should appear on the calendar for the community centre, viewable by staff and other users of the site. However, only administrators should see the personal details of the user that made the booking. (see figure 1)
Exception handling	Customer can abandon their booking before paying - If they haven't yet paid, the system needs to ignore the booking. Concurrent bookings - Concurrency control to ensure that no-one ends up paying for a booking that ends up clashing. Deal with this using RTTS. This means that when anyone is paying for a booking, other users are locked out from booking in their time slot. This stops any issues of double payment.

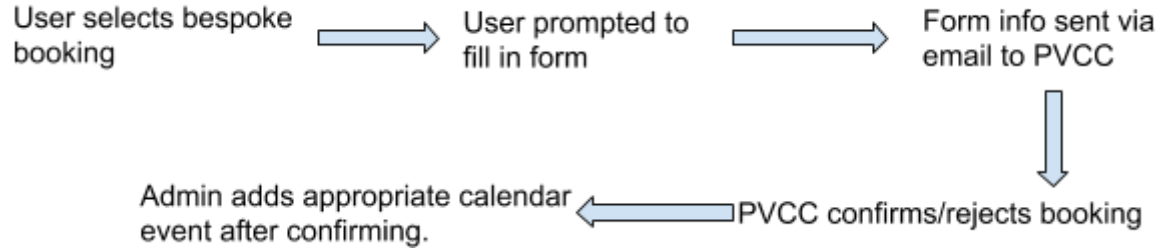
ID, Title	FR1.2 - Payment system
Description	Customers should be able to pay securely by card for facilities. Payments should be refundable up until 24 hours beforehand provided the customer contacts PVCC and provides the payment reference.
Dependencies	FR1.1, FR2, FR4
Priority	High
Expected Results	Once the customer has made a request on the online booking system and is prompted to pay, they should be able to enter their card details on a secure page, and make a payment. If successful, the customer should be redirected to the page and be able to view their booking. (see figure 1)
Exception Handling	If the customer closes the payment window early,or an error occurs during payment, the site shouldn't register their payment as having gone through.

Figure 1 - Booking (FR1.1) and Payment (FR1.2) System Flow Diagram



ID - Title	FR1.3 - Online Non-Automated Booking Requests
Description	For booking bespoke functions which use multiple facilities or a variable amount of time, a non automated booking request should be able to be sent through a form on the website. Also, if customers don't want to pay online, they should be able to book via this system These requests will be manually processed by PVCC.
Dependencies	FR4
Priority	High
Expected Results	User should be able to enter their contact details and booking request into a form on the website. Submitting the form should automatically send an email to PVCC employees with this information. Employees should be able to add the function to the PVCC calendar manually. (see figure 2)
Exception handling	Concurrency issue with employee entering information into the calendar, and an automated booking attempting to access the calendar. RTTS(see FR1.1 exception handling) should prevent the user booking via the automated system from booking (and most importantly paying).

Figure 2 - Non Automated Booking System (FR1.3) Flow Diagram



ID, Title	FR2 Calendar Synchronisation
Description	Synchronise PVCC's calendar and the calendar of Park View school. (visible on http://parkviewlearning.net/our-school/dates/school-calendar/) This way, users on the PVCC site cannot book over school activities.
Dependencies	None
Priority	High
Expected Results	Activities on the Park View School calendar should automatically appear on the PVCC calendar. School Activities that take up certain facilities should prevent automated booking from taking place.
Exception Handling	If the school adds a new activity in a time slot to the calendar, it shouldn't overwrite a customer's booking that is already in our calendar. This situation would have to be flagged and resolved by a PVCC employee manually contacting the school and the user who made the booking. PVS may put new activities on their calendar where we don't know what facilities will be used. In this case, users should still be able to send a non automated booking request, which can be resolved manually, but the automated system should treat the booking as an overlap.

ID, Title	FR3.1 - Website redesign
Description	Change the layout of the website to accommodate the new booking and payment systems.(FR1) Improve the design of existing parts of the website, such as the price list, and the link to “easyfundraising”. More design improvements are listed in requirements 3.2 - 3.4
Dependencies	FR1.1, FR1.3
Priority	Medium
Expected Results	Users of the site should be able to book facilities with an easily visible link from the facility pages. The price list of the facilities should be easier to find for customers looking to make bookings. The “easyfundraising” link should not take up so much space on the website’s homepage

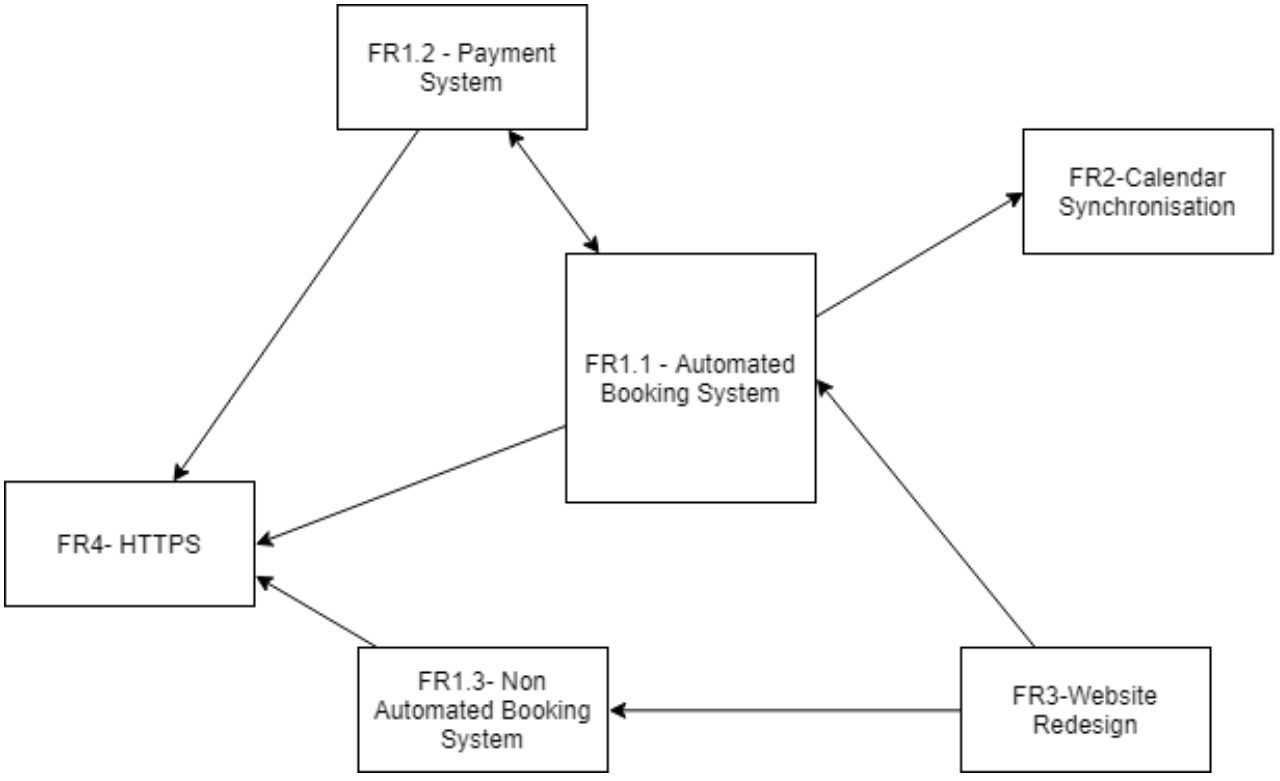
ID, Title	FR3.2 - Twitter Feed
Description	As of now, the website lacks a proper Twitter feed, all they have is a little Twitter icon. We are hoping to upgrade this to a proper feed where people on the website can see the most recent events.
Dependencies	None
Priority	Medium
Expected Results	The users should be able to see the top 3 most recent PVCC posts (at least). The feed will automatically update whenever a post is made.

ID, Title	FR3.3 - Fix Google Maps link
Description	The Google Maps display of PVCC's location on the 'contact us' page is broken. We aim to fix this functionality.
Dependencies	None
Priority	Medium
Expected Results	Visitors should see the two locations of PVCC pinpointed on the Google map when they visit the 'contact us' page.

ID, Title	FR3.4 - 'What's on' page and homepage cards
Description	The 'what's on' page is a table of activities, currently the table is partially broken, with content tags visible in parts of the table instead of contents. 'What's on' cards showing activities appear on the homepage, currently, these seem to be static, rather than updating dynamically based on the 'what's on' page. We aim to improve both the table, and the homepage cards.
Dependencies	None
Priority	Medium
Expected Results	'What's on' page table should be fixed, so that content tags are no longer visible in parts of the table 'What's on' cards on the homepage should be updated based on the current day of the week, and the corresponding part of the 'what's on' table.

ID, Title	FR4 - HTTP to HTTPS
Description	The website currently uses HTTP however, if it is to be handling online payments and personal data (email), we will need to transfer it into HTTPS to secure the website. This adds a secure socket layer (SSL) to all data being sent to/from the website, encrypting it.
Dependencies	None
Priority	High
Expected Results	All form/payment data should be sent encrypted. Website address changed to https://www.pvcc.co.uk.
Exception Handling	The current host may not support an SSL certificate, which is needed for switching the site to HTTPS. All links on the site must be switched to use a secure protocol, otherwise a browser may give mixed content warnings [4] about an HTTPS site that has non secure links. This can be handled by carefully scanning resources the site uses, and switching their links to the HTTPS versions.

Figure 3 - Functional Requirements Dependencies Diagram



Non functional requirements

Security	Only PVCC admins will have the privilege to edit content on the site such as event descriptions and the calendar. The website will be handling online payments, which means sensitive data may be passing through the site. Hence, it must be ensured that this data remains private. The website should be resilient to common attacks, such as cross site scripting.
Reliability	The website will be available at all times. Manual requests will not be handled outside of the standard PVCC working hours.
Usability	Website - The user should be able to navigate to any page of the website within 2 clicks (preferably 1 click), and find all of the information relevant to their search on a single page. For example, if you click on a facility, the price should be displayed, and a link to booking/paying for the facility should be clearly visible. Admin page - The client lacks technical expertise which means that the existing Wordpress administration page must be modified[5] such that it is simple for the admins to understand. They should know at a glance what each button does and the system should flow well into their natural intuition.
Accessibility	The devices that the website could be viewed from are varied. We will have to ensure the website is fully functional with a touch screen and on a variety of screen sizes.
Scalability	During busy times on the website, concurrent bookings will be handled appropriately (refer to FR1.1 exception handling)
Legality	The website must comply with the GDPR regulation. User data must remain private. The booker must consent to having their data handled by PVCC. PVCC must have a designated data controller to handle requests relating to GDPR like Right to Access.
Longevity	The client should be able to make changes to the site without our help and keep it updated. (see admin page, usability)

Open Issues

Risks

Contacting client [High Risk]:

- Currently we can only contact our client via email. In the days preceding the writing of this document, the client has been unresponsive. Unpredictable response times could cause problems with certain deadlines that require resources from the client, for instance the colour palette for their theme and their Wordpress account for editing the website.

Site creator is currently unreachable [Low Risk]:

- The person who originally built the website, a former student working in a company, is currently unable to be contacted. Since this person has access to all the internals of the site, failure to contact him poses a small risk.
- The risk is only small because a member of PVCC allegedly has access to the site, which would remove the need to contact this person.

Website Redesign features (Google Maps, Google Calendar, Twitter) [Moderate Risk]:

- Features such as Google Calendar, Google Maps and Twitter feeds rely on their respective 3rd party APIs. In the long term, it can be expected that the APIs will get updated which then runs the risk of breaking compatibility.
- The highest priority issue is the Google Calendar API, the booking system completely hinges on this, and has to hinge on this. If an update on Google's end breaks the calendar, then the functionality of the booking system could be compromised.
- However, updates of this scope are few and far between on Google API's so the website is likely to maintain functionality for a long time. Therefore, we decided on a moderate risk level.
- Google Maps and Twitter are lower priority features, but are still liable to break in the same way.
- With regular maintenance of the website, it should be possible to prevent these features from going down.

Restraints/Limitations

GDPR compliance:

- Since we will be handling personally identifying information such as emails, we need to ensure that this information is not compromised, intentionally or unintentionally.

Bespoke requests must be handled manually - True automation most likely impossible:

- A user may request multiple facilities for unusual activities that PVCC may or may not be able to provide. Phone calls are necessary.

Google Calendar will have be used for syncing between sites:

- This is because the school website uses Google Calendar. We cannot change the system on that end, limiting us to Google Calendar API.

Wordpress is going to be used for development and maintenance in the future:

- This is the tool the website uses currently. Hence, it is the tool the clients are familiar with. As our client has stated their lack of technical expertise, we should avoid changing up the system to make it easier on them.
- A potential alternative would be to get all the source code for the website and develop the site offline with our own editors. However, we would still have to make sure that our code is compatible with Wordpress.

Development Approach

There are many development approaches available, from a structured approach like the Waterfall Model to an agile one like XP. These are approaches only and we are not following any method to the letter. This would just be restrictive and for a first-time group project we need to be able to adapt to what works.

Context and Justification

We are building a website and booking system - a small-scale project. For most of us it will be the first time working collaboratively, and we have mandated prototypes which forces us to use an iterative approach. It is worth noting that none of our prototypes will go live, therefore they do not necessarily need full functionality. We also need a development approach which allows us to design as we tackle each feature, respond well to changing client requirements, have constant communication and clarification with the client, and for team members to work to their strengths in different areas of the project.

Approaches Considered

We are discarding any rigidly structured approach, such as the Waterfall Model, categorised by distinct and linear stages of design, planning, development, and testing. This will naturally not work as it has been stipulated that prototypes must be produced, necessitating some iterative method. As well as this, a structured approach is best suited to a large system in which the requirements are well defined from the start, because this is when planning occurs. The client for this project has expressed that the requirements may change as development occurs; more may be added if it is possible in the timeframe, and we may discover new requirements, or opportunities for further development, as we begin to develop parts of the solution.

The whole team would need to be focussed on each stage at the same time, and while this may be a good thing in some sense as we would all be thinking about the same problem, it also means that if one member were not as good at a certain aspect of development like design, their contribution would not be as valued. A better approach would allow them to work on another development aspect, allowing them to utilise their skills more appropriately.

Our Approach

We will use a Prototyping Process Model. Unlike Agile, which focuses on each iteration developing a specific feature to completion, prototyping features a more breadth first approach - in which the whole system is mocked up and slowly implemented. The first prototype will focus on design over functionality with a range of templates and ideas to present so we can gather useful feedback from the client. The implementation details are not relevant to the client, so these can be done later once we have a good understanding of the system they envision.

Once we are confident the client will be happy with the implementation, we will take a more agile approach, using iterative development to add features taking a priority first approach. We will meet at least weekly to discuss progress, plans and problems. At the production of each prototype we will present our work to the client to gather feedback and identify new priorities that will feed into the next prototype. We are going to try and maintain a continuous

email dialogue with the client to clarify any points of dispute, like how pricing will be handled and collecting resources for development like the colour palette. We need to consciously avoid negative compromises such as not fulfilling a requirement late in development due to time constraints, forcing an incomplete or unrefined product onto the PVCC website; therefore, time management and delegation of tasks will be a concern. An approach involving the client heavily will benefit them as they will be able to influence development throughout and get an end system more tailored to their vision.

Testing Strategy

Individual sections of code and components will be tested thoroughly through unit testing to ensure they are robust and do what is intended. Where possible, tests will be written before the implementation to avoid bias. An example of something that will be unit tested is a button on the website. The next level is integration testing, done on features built of individual components such as the booking system. Then there is the overall system testing that verifies holistic functionality.

Non-functional requirements will be tested less precisely. For example, we will ask people to use the website and give feedback as well as using browser developer tools to emulate other devices to test usability and compatibility. We will do penetration testing on a local version to verify security, and take measurements to analyse performance.

Hardware, Software, Technologies and Version Control

The website will remain cloud hosted, so the server side hardware is determined by the host and we don't need to worry about it.

There are several technologies we are planning on using for this project. Namely, we are planning on using the Google Calendar API for integration of PVS and PVCC calendars. The website is currently built on Wordpress, which itself is developed with PHP; we plan on continuing to use this to modify the site. However, as with any website, there will also be the standard web technologies of HTML, CSS, and JavaScript for client-side scripting.

We will use PayPal to handle online payments; this will let us handle card details indirectly, reducing security risk. PayPal is a good choice because it is widely used, so likely to be familiar to users.

We will be using Git for version control as it will allow us to keep track of all changes and the sources of contribution. Git branches allow multiple development lines at once without conflict, which is ideal for a group project. We will also use GitHub to store the project centrally so that the working version is cloud hosted, allowing us to push our own changes to the repository at any time. GitHub also has many project visualisation tools which will be useful, such as the ability to see a graphical representation of all branches and commits.

Terminology

Term	Meaning
PVCC	Park View Community Centre
PVS	Park View School
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
API	Application Programming Interface
FR	Functional Requirement
PHP	Hypertext PreProcessor [Programming language]
RTTS	Real-Time Transaction System
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
SSL	Secure Socket Layer
GDPR	General Data Protection Regulation
UX	User eXperience
XP	Extreme Programming

Project Schedule

	15 Oct	22 Oct	29 Oct	5 Nov	12 Nov	19 Nov	26 Nov	3 Dec	10 Dec	17 Dec	24 Dec	31 Dec
Requirement Specification												
First Prototype				S0		S1		S2				
Second Prototype								S3		S4		
	7 Jan	14 Jan	21 Jan	28 Jan	4 Feb	11 Feb	18 Feb	25 Feb	4 Mar	11 Mar	18 Mar	25 Mar
Second Prototype	S4											
Final Product					S5				S6			
Software Report												
Test Plan Report												

Prototype 1

The primary stage of this prototyping phase is to complete designs and mockups of the website. These will be presented to the client for feedback. We can also start researching how to technically implement our design ideas.

Soft deadlines:

S0 - Acquiring resources from client

S1 - Design mockups

S2 - Implementation research

Prototype 2

After receiving client feedback, the second stage of prototyping will involve making any necessary design changes, and then focusing more heavily on implementation.

We should aim to have at least a partially working system by the end of this phase.

Soft deadlines:

S3- Design changes

S4 - Working prototype implementation

Final Product

After completing the partial implementation from the second phase of prototyping, the system should undergo final testing, bug-fixing and review before delivery.

Soft deadlines:

S5 - Finishing implementation

S6 - Final testing and review

References

[1]-https://www.thetrainline.com/ Trainline, accessed 25/10/18

[2]-https://www.dur.ac.uk/library/password/rooms/ (*login required) Durham University Library room booking system, accessed 25/10/18

[3]-https://www.hss.com/hire HSS Hire, accessed 01/11/18

[4]-https://developers.google.com/web/fundamentals/security/prevent-mixed-content/what-is-mixed-content - Mixed content warnings in HTTPS, accessed 15/11/18

[5]-https://www.codeinwp.com/blog/how-to-simplify-the-wp-admin/ - Making the existing Wordpress admin page simpler for clients to view, accessed 15/11/18