

comp90082-2024-sg-koala3
BACKGROUND7
Product Overview8
Stakeholders9
Client Expectation11
Industry Need12
Scopes13
Terminology15
Choice of Technology16
Constraints17
Ethical Consideration18
Cyber Security Consideration19
Deployment Decision20
ROADMAP22
High-Level Plan23
SPRINT-124
Sprint-1 Plan25
Sprint-1 Goal26
Sprint-1 Burn Down Chart27
Sprint-1 Review28
SPRINT-230
Sprint-2 Plan31
Sprint-2 Goal32
Sprint-2 Burn Down Chart33
Sprint-2 Review34
SPRINT-335
Sprint-3 Plan36
Sprint-3 Goal37
Sprint-3 Burn Down Chart38
Sprint-3 Review39
SPRINT-440
Sprint-4 Plan41
Sprint-4 Goal42
Sprint-4 Burn Down Chart43
Sprint-4 Review44
REQUIREMENTS45
Goal Model46
Personas47
Prototypes49
User Stories - Product Backlog50
Provided Information - user stories support details59
RISK61
Risk Registry62
Review Meetings63
COLLABORATION64

About Us	65
Team Calendar	66
Agile Processes	67
Team Charter	68
Definition of Ready	70
Definition of Done	71
Agile Ceremonies	72
Sprint Planning - 1	73
Retrospective - 1	75
Review Meeting - 1	77
Sprint Planning - 2	79
Sprint Planning - 3	80
ARCHITECTURE	82
Database Model	83
API	85
TESTS	99
Backend - Testing	101
Frontend - Testing	103
CODE REVIEW	104
Frontend Code Review	105
Backend Code Review	111
Front-end check list before doing the merge request	114
Back-end check list before doing the merge request	115
LINKS & RESOURCES	116
Demo	117



comp90082-2024-sg-koala

About

The project background for the CIS Booking System revolves around the needs of the Museums & Collections (M&C) Learning Team at the University Of Melbourne.

The existing booking system faces challenges due to the complexity of managing multiple venues, varying spaces within these venues, different staff rosters, diverse booking processes, and invoicing methods. Schools show interest in these programs through various channels, including direct negotiations, online bookings for self-guided tours, Microsoft Forms for workshops, and general inquiries via phone or email.

The process involves multiple steps such as recording requests, checking venue availability, managing staff schedules, confirming details with schools, booking buses if needed, issuing invoices for non-partner schools, and handling additional logistical requests. The system currently utilizes tools like Planner, Priava, Microsoft Calendar, and Excel to manage these tasks, but it lacks integration and automation, leading to potential inefficiencies and inaccuracies in booking management.

The intent to upgrade to a new system arises from the need for a more cohesive and efficient approach to managing these complexities. The new system aims to streamline the booking process, enhance organizational efficiency, and improve the overall management of educational excursions by addressing the existing system's limitations and incorporating a more integrated, user-friendly solution.

Status

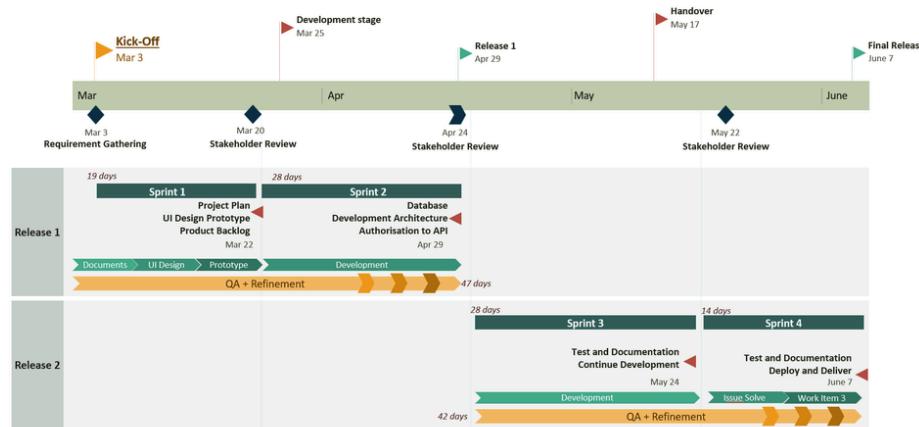
We are working on this!

[Expand all](#) [Collapse all](#)

- ✓ **BACKGROUND**
 - Product Overview
 - Stakeholders
 - Client Expectation
 - Industry Need
 - Scopes
 - Terminology
 - Choice of Technology
 - Constraints
 - Ethical Consideration
 - Cyber Security Consideration
 - Deployment Decision
- ✓ **ROADMAP**
 - High-Level Plan
- ✓ **SPRINT-1**
 - Sprint-1 Plan
 - Sprint-1 Goal
 - Sprint-1 Burn Down Chart
 - Sprint-1 Review
- ✓ **SPRINT-2**
 - Sprint-2 Plan
 - Sprint-2 Goal
 - Sprint-2 Burn Down Chart
 - Sprint-2 Review
- ✓ **SPRINT-3**
 - Sprint-3 Plan
 - Sprint-3 Goal
 - Sprint-3 Burn Down Chart
 - Sprint-3 Review
- ✓ **SPRINT-4**
 - Sprint-4 Plan
 - Sprint-4 Goal
 - Sprint-4 Burn Down Chart
 - Sprint-4 Review
- ✓ **REQUIREMENTS**
 - Goal Model
 - Personas
 - Prototypes
 - User Stories - Product Backlog
 - Provided Information - user stories support details
- ✓ **RISK**
 - Risk Registry
 - Review Meetings
- ✓ **MEETINGS**
 - The 1st Client Meeting
 - ✓ **Team Meetings**
 - Team Meeting - 0
 - Team Meeting - 1
 - Team Meeting - 2
 - ✓ **Supervisor Meetings**
 - Supervisor Meeting - 0
 - Supervisor Meeting - 1
 - Supervisor Meeting - 2
 - Supervisor Meeting - 3
 - Supervisor Meeting - 4
 - Supervisor Meeting - 5
 - Supervisor Meeting - 6
 - Supervisor Meeting - 7
- ✓ **COLLABORATION**
 - About Us
 - Team Calendar
 - ✓ **Agile Processes**
 - Team Charter
 - Definition of Ready
 - Definition of Done
 - ✓ **Agile Ceremonies**
 - Sprint Planning - 1
 - Retrospective - 1
 - Review Meeting - 1
 - Sprint Planning - 2

- * Sprint Planning - 3
- ✓ ARCHITECTURE
 - * Database Model
 - * API
- ✓ TESTS
 - ✓ Backend - Testing
 - * Unit Tests - GET
 - * Unit Tests - PUT
 - * Unit Tests - DELETE
 - * Unit Tests - CREATE
 - ✓ Frontend - Testing
 - * Unit Test - Dashboard
 - * Unit Test - NewBooking
 - * Unit Test - Templates
 - * Unit Test - TemplateDetail
- ✓ CODE REVIEW
 - * Frontend Code Review
 - * Backend Code Review
 - * Front-end check list before doing the merge request
 - * Back-end check list before doing the merge request
- ✓ TEMPLATE
 - * Template - Decision documentation
 - * Template - Product requirements
 - * Template - Meeting notes
- * LINKS & RESOURCES
 - * Demo

Roadmap





BACKGROUND

1. Product Overview:

- The **core functionality** and **purpose** of the product.
- How the product **addresses specific problems**.
- **Key features** of the product.

2. Stakeholders:

- Lists **key stakeholders** involved in the project, including both internal and external parties.
- Describes the **role, level of influence, impact** and **communication plan** of each stakeholder on the project.

3. Client Expectation:

- Specific **requirements** and **expectations** of the client regarding the product.

4. Industry Need:

- How the **product fits** into **specific client needs**.

5. Scopes:

- Defines the **scope of the project**.
- Project **objectives** and **functionalities**.

6. Terminology:

- Professional **terms** and **definitions** used in the project, and product.

7. Choice of Technology:

- Reasons for selecting **specific technologies or platforms**, including software, hardware, development tools, and frameworks.
- Technical **dependencies** and integration needs.

8. Constraints:

- Limitations faced by the project.

9. Ethical Consideration:

- Consideration of **issues and decisions** in the project

10. Cyber Security Consideration:

- Consideration of **security issues** in the project

11. Deployment Decision:

- **Deployment decision** of backend and frontend

Product Overview

Introduction:

Science Gallery Melbourne introduces multiple unique tours and workshops for secondary school group visits. With a high volume of booking requests throughout the year, the current use of multiple systems including Microsoft Office tools and Priava for space and time management has become cumbersome.

The primary challenge is the complexity and inefficiency of using multiple systems to manage bookings. This multi-step process requires constant updates across several platforms when any changes to bookings are made, involving a significant administrative effort and coordination among team members.

We seek to develop a Unified Booking Management System that simplifies and automates the booking process. This solution will integrate booking, calendar scheduling, and space and equipment management into a single, user-friendly platform. It will help the staff of the Learning Team add and update bookings quickly, check for any scheduling conflicts and store all the tours to the database automatically.

Key Features:

1. **Unified Booking System:** A centralised platform that integrates booking requests, calendar management, and resource allocation, ensuring a seamless process from initial inquiry to final confirmation.
2. **Dynamic Scheduling & Resource Management:** Checks for space, staff availability, and equipment requirements to prevent double bookings and ensure that all necessary resources are allocated to each event.
3. **Bulk Editing & Updating:** Offers the capability to modify and update multiple bookings simultaneously, facilitating easy adjustments to recurring events or changes in program schedules.
4. **Comprehensive Database Integration:** Confirmed and completed bookings are systematically recorded in a database, enabling efficient tracking, reporting, and analysis of past events for continuous improvement and planning.
5. **Interactive Weekly Dashboard:** Provides a real-time overview of upcoming programs, including detailed information on the schedule, assigned staff, and preparation requirements, aiding in efficient staff rostering and program delivery.

Benefits:

Efficiency: Automates and simplifies the booking and event management process, saving time and reducing manual errors.

Flexibility: Easily accommodates various program types, booking requirements, and resource constraints.

Reduce Human Error: Minimises the likelihood of mistakes in booking arrangements and resource allocations, ensuring accuracy and reliability in every event organised.

Stakeholders

Stakeholders	Description	Level of Interest	Level of Influence	potential Impact	Communication Plan
Science Gallery Learning Team	Main end user of the product, responsible for organising and managing educational programs and workshops at the Science Gallery Melbourne.	High, as the product directly affects their work processes and efficiency.	High, given their direct involvement in defining requirements and using the system.	High, as improvements in the system can greatly enhance their operational efficiency and program quality.	In person meeting after each sprint.
Partner/Non-partner school teachers	Teachers from schools that collaborate with the gallery for educational tours and workshops.	Low	Low	Low, their booking process will not change.	N/A
Project team	The group of individuals tasked with developing and implementing the project.	High, as their role is to deliver a successful product.	Very high, as they make the crucial decisions on the system's features and functionality.	High, as the project's success directly correlates with their performance.	Meet once or twice a week in person or through zoom.
University of Melbourne	The university associated with the Science Gallery Melbourne, potentially providing resources or support.	Moderate	High, the booking process need to use specific tool.	Moderate	N/A
Learning team temporary employee	Short-term staff hired to assist with the increased workload or specific tasks within the Learning Team.	Moderate	Low	Moderate, in terms of testing and providing feedback on the system's usability and efficiency.	N/A
Bus company BusCharter	The service provider responsible for transporting students to and from the gallery.	Low	Low	Low	N/A
Interpreters	The service provider responsible for translating for students	Low	Low	Low	N/A

Unimelb business service	The business services division within the University of Melbourne that responsible for paying additional fees (bus and interpreters).	Moderate , as they need accurate information sent from the system	Moderate	Moderate	N/A
Secondary school students	Students that participate in tours and workshops.	Low	Low	Low	N/A
IT department	The team responsible for maintaining the technological infrastructure.	High, as they need to ensure the system's reliability and security.	High, due to their role in troubleshooting, maintenance, and potentially implementing technical improvements.	High, as the system's functionality and user experience heavily depend on their efficiency and effectiveness in addressing technical issues.	Communicate using email.

Client Expectation

1. **Unified Booking System:** A centralised platform that integrates booking requests, calendar management, and resource allocation, ensuring a seamless process from initial inquiry to final confirmation.
2. **Dynamic Scheduling & Resource Management:** Automatically checks for staff availability, and equipment requirements to prevent double bookings and ensure that all necessary resources are allocated to each event.
3. **Bulk Editing & Updating:** Offers the capability to modify and update multiple bookings simultaneously, facilitating easy adjustments to recurring events or changes in program schedules.
4. **Comprehensive Database Integration:** Confirmed and completed bookings are systematically recorded in a database, enabling efficient tracking, reporting, and analysis of past events for continuous improvement and planning.
5. **Interactive Weekly Dashboard:** Provides a real-time overview of upcoming programs, including detailed information on the schedule, assigned staff, and preparation requirements, aiding in efficient staff rostering and program delivery.
6. **User-Friendly Interface:** The system should be intuitive and easy to navigate for all users.

Industry Need

The project, while focused on enhancing the booking and event management processes for the Science Gallery Melbourne, taps into a broader industry need within large educational and cultural institutions like the University of Melbourne. The specific challenges related to efficiency in such expansive organisations underscore a critical industry demand. These challenges are often compounded by the size and complexity of the organisation, leading to inefficiencies in communication, coordination, and resource management.

Scopes

Inner Scope	Form for new bookings	<p>Objectives: A form for staff to enter booking details when clients book via phone or email.</p>
		<p>Functionality: The form should contain all the necessary details and specific request. A booking ID will be sent to client email after form is completed.</p>
	Tag pending application	<p>Objectives: Tag pending application depends on the programs applied.</p>
		<p>Functionality: Staff will be able to tag pending application to generate checklist using different template.</p>
	Checklist	<p>Objectives: A checklist shows following steps that need to be complete.</p>
		<p>Functionality: Staff will use the checklist to ensure all necessary steps are taken, such as availability confirmation, payment processing, final confirmation with clients.</p>
	Calendar	<p>Objectives: A calendar shows all the confirmed bookings.</p>
		<p>Functionality: Confirmed bookings will be visible on calendar, with accompanying staff and number of students.</p>
	Database	<p>Objectives: A database will store all details of the bookings.</p>
		<p>Functionality: Staff will be able to add or edit new data. Generates analytic reports for future reporting.</p>
Outer scope	Secondary school application form	<p>Partner/Non-partner school teacher apply tour/workshop by filling online form.</p>

	Priava venue	Staffs check and book venue on Priava.
	Microsoft Calendar events	Add or update events on Microsoft Calendar, accompanying teacher will be assign to events.
	Payments	Pay necessary fees such as bus or interpreter fees.

Terminology

Term	Definition
Booking System	A digital platform used to manage reservations, appointments, or event participations, allowing users to schedule visits or services in advance.
Resource Allocation	The process of assigning available resources in an efficient, effective, and orderly manner. In the context of this project, resources can include staff, rooms/spaces, equipment, and workshop materials.
Dashboard	A visual interface that displays key information and metrics at a glance, allowing users to monitor and manage activities, resources, or performance indicators effectively.
Database	A structured set of data held in a computer, especially one that is accessible in various ways.

Choice of Technology

Choice of Technology	
Database	MongoDB
Front-End	React Framework
Back-End	Django Python

Client Requirement : Any technology is feasible, as long as it does not affect priava's calendar.

- **Database:** MongoDB

- what: MongoDB is an open-source NoSQL database that stores data in the form of documents (usually in JSON or BSON format). Compared to traditional relational databases (such as MySQL or PostgreSQL), it offers higher flexibility and scalability.
- why: For this project, given the limited budget, using the free version of MongoDB is a relatively good choice, as it can largely meet our requirements.

- **Front-End:** React Framework

- what: React is an open source plug-in JavaScript library for building user interfaces. Since It has become one of the most popular and widely used tools in plugin development. The main goal of React is to improve development efficiency, enable fast UI rendering, and help developers build large, dynamic web applications.
- why: Our team members are essentially experienced with React, which makes it very easy for us to get started. We can begin developing our website earlier, saving some time for learning and adaptation.

- **Back-End:** Django Python

- what: Django is an advanced Python web framework designed to make developing complex, database-driven websites easier and faster.
- why: Django's comprehensiveness can cover all our needs, and some of our team members also have rich experience using it, so we choose to use Django.

Constraints

- This product is required to use the Priava booking system for venue booking
- This product is required to use Outlook Calendar to invite facilitators and manage booking calendar
- Zero budget on deployment, database

Ethical Consideration

Privacy and Data Security:

- **Personal Information:** Ensuring that personal information provided by users is protected from unauthorised access and misuse is crucial. The system should comply with relevant data protection laws and use secure data encryption methods.
- **Access Controls:** Ensure that only authorised staff have access to sensitive customer data and that access levels are appropriately assigned based on job roles.
- **Data Handling:** Training staff on secure data handling practices is essential, as human error is a common security vulnerability. Additionally, the organisation should establish a robust data breach response plan to quickly address and mitigate the impact of data leaks.

Transparency and Honesty:

- **Error Reporting:** Systems should be transparent in terms of functionality and reporting, enabling staff to easily report any issues or inaccuracies in bookings. This system should be easy to use and integrated into the daily operations, ensuring that errors are corrected before they can do any serious impacts.
- **Communication:** Policies should be in place to ensure that all staff members communicate consistently and openly with customers, especially regarding cancellations, changes, or errors. This transparency builds trust and prevents misunderstandings. Training should emphasise the importance of clear and honest communication, reinforcing the company's commitment to customer service.

User Friendliness and Accessibility

- **Interface Design:** The design of the booking system interface should prioritise simplicity and efficiency, reducing the potential for user error and enhancing the overall user experience for staff such as clearly labeled functions and a logical navigation layout.
- **Accessibility Features:** Consider features that make the system accessible to staff with disabilities, such as screen readers or alternative input methods.

Technical Support

- **Responsive Technical Support:** A dedicated technical support team should be available to address system issues swiftly, minimising downtime and ensuring that the booking process is smooth for staff. This support should be proactive, identifying potential problems before they affect system performance.

Cyber Security Consideration

Secure Data Transmission and Storage:

Strong encryption protocols should be implemented for data at rest and in transit. For data in transit, TLS (Transport Layer Security) should be standard to secure communications between users' browsers and the server. For data at rest, robust encryption algorithms should be used to protect it from unauthorised access if the data storage is breached.

Access Controls:

Ensure that only authorised staff have access to sensitive customer data and that access levels are appropriately assigned based on job roles.

Security Audits:

Conduct regular security audits to assess the effectiveness of the security measures in place. These audits should check for vulnerabilities in both software and hardware components of the booking system.

Penetration Testing:

Employ ethical hackers to perform penetration testing. This involves attempting to exploit system vulnerabilities to determine where the system can be fortified. It helps identify weaknesses before they can be exploited by malicious actors.

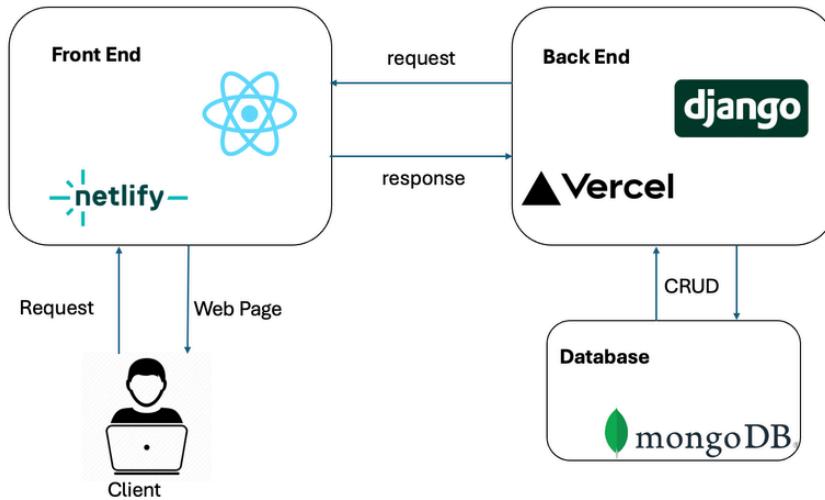
Regular Backups:

Regularly back up data and ensure that backups are stored in a secure location. Backups can be vital in recovering from a data breach, ransomware attack, or other types of data loss incidents.

Regular Updates:

Implement a strict schedule for updating all software, including the operating system, applications, and all third-party software components used in the booking system. This helps protect against vulnerabilities that could be exploited by attackers.

Deployment Decision



Deployment Tool Options:

1. Netlify:

- **Description:** Netlify is a popular hosting and automation platform for static websites.
- **Pros:** Seamless integration with Git, continuous deployment, serverless functions, and CDN.
- **Cons:** Limited support for backend services compared to full-fledged cloud providers.

2. Vercel:

- **Description:** Vercel is a cloud platform for static sites and serverless functions, with a focus on speed and simplicity.
- **Pros:** Built-in support for serverless functions, CDN, and seamless integration with Git.
- **Cons:** Limited scalability and flexibility compared to full cloud providers like AWS or Azure.

Evaluation Criteria:

- **Ease of Use:** How easy is it to set up and use the tool?
- **Compatibility:** Does the tool support your technology stack and deployment environment?
- **Scalability:** Can the tool handle your current deployment needs and scale with your project?
- **Community Support:** Is there a strong community around the tool for support and updates?
- **Cost:** What is the cost of the tool and is it within your budget?
- **Features:** What specific features does the tool offer that are beneficial for your project?

Decision Criteria:

- **Technical Fit:** Do the tools meet your technical requirements for frontend and backend deployment?
- **Team Expertise:** Does your team have experience or expertise with Netlify and Vercel?
- **Cost-Effectiveness:** Are the costs of using both tools justified by their features and benefits?
- **Scalability:** Can Netlify and Vercel scale to meet your future deployment needs?
- **Support and Maintenance:** Is there adequate support and maintenance available for Netlify and Vercel?

Decision:

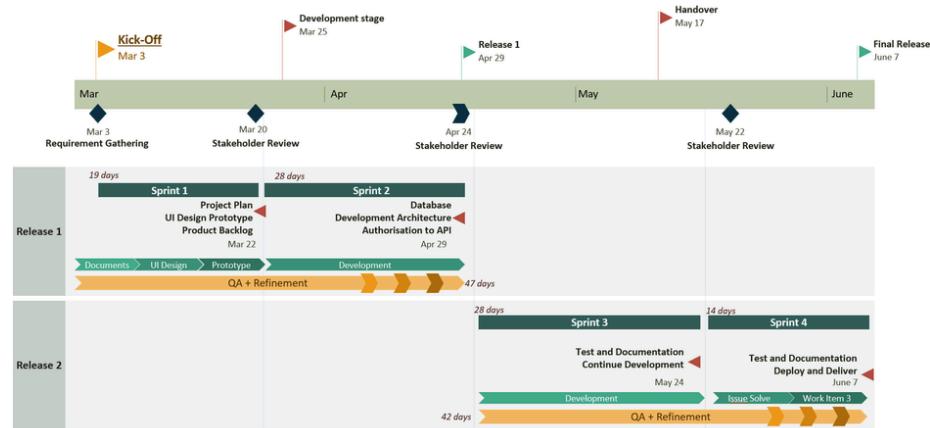
- **Selected Frontend Tool:** Netlify
- **Selected Backend Tool:** Vercel
- **Reasons for Selection:** Both Netlify and Vercel have free tier that the team can use in the development stage

Risks and Mitigation:

- **Risk:** Potential compatibility issues between Netlify and Vercel.
- **Mitigation:** Thorough testing and coordination between frontend and backend teams.

Stakeholder Communication:

- **Communication Plan:** Notify client about the decision to use Netlify and Vercel for frontend and backend hosting.

 ROADMAP


High-Level Plan

Sprint 1 - planning

- design prototype
- determine the choice of technologies
- generate product backlog
- estimate story points and priorities
- allocate user stories for sprint 2
- allocate development roles
- generate risk management strategy

Sprint 2 - development

- client review on sprint 1
- update requirements based on client review
- Initialise development architecture, front end and back end
- design database structure
- enable authorisation to external API
- complete user stories allocated
- monitor risk
- update risk management strategy if detect new risk
- update development speed

sprint 3 - development

- client review on sprint 2
- update requirements based on client review
- continue development on allocated user stories
- update risk management strategy if detect new risk
- test and document completed features
- complete user stories allocated
- update development speed

sprint 4 - deployment

- test and document all features
- client review on sprint 3
- final adjustment based on client review
- deploy and deliver

1

SPRINT-1

Sprint-1 Plan

- design prototype
- determine the choice of technologies
- generate product backlog
- estimate story points and priorities
- allocate user stories for sprint 2
- allocate development roles
- generate risk management strategy

Sprint-1 Goal

- collect client information for user stories
- generate product backlog
- generate sprint 2 goals and burn down chart
- generate goal model
- confirm the choice of technologies
- confirm communication frequency
- initialised risk management strategy
- complete design prototype

Sprint-1 Burn Down Chart

Since Sprint 1 is the inception and design phase, there is no need for a burn-down chart of user stories.

Sprint-1 Review

Date & Time

09:30 Mar. 27th, 2024

Participants:

- @Guixian Li
- @Yun-Chi Hsiao
- @Yongli Qin
- @Pangfeng ZHENG
- @Daniel Su
- @Yuanbo Xu
- The client

Product Demonstration:

Features Demonstrated:

Figma review of design: The client reviewed the current design, appreciating its overall quality but suggested specific improvements:

- Enable the search bar to find both school and non-school partners.
- Include a textbox for staff to paste booking details from emails.

Client Feedback:

Key point Discussed:

- Chris will provide the email template.
- When clicking a hyperlink, it will open in a new page rather than being embedded in the current page.
- Consideration of whether different clients should use the same account.

Decision made:

- All users can access the same account from multiple devices.
- Integration of hyperlinks in Priava, Microsoft, and the bus booking system should open in a new page.
- Addition of an email textbox in the form creation process

Review of the Sprint:

Achievements:

- Successfully collected comprehensive client information necessary for creating detailed and accurate user stories.
- Generated a well-structured product backlog that prioritises features, tasks, and requirements identified from the collected user stories.
- Clearly defined and communicated goals for Sprint 2, developed a burn down chart.
- Successfully completed and demonstrated the design prototype, allowing the team and client to visualise the end product and facilitate early testing of concepts and interfaces.
- Finalised and confirmed the choice of technologies to be used in the project.

Action Items

- Review and update Figma designs to include new search bar capabilities and other client's needs. @Yongli Qin @Yun-Chi Hsiao
@Guixian Li
- Review and improve the structure of project documentation on Confluence. @Yuanbo Xu @Daniel Su @Pangfeng ZHENG
- Update Trello to reflect changes in user stories and tasks based on today's feedback. @Pangfeng ZHENG

2 SPRINT-2

Sprint-2 Plan

- client review on sprint 1
- update requirements based on client review
- Initialise development architecture, front end and back end
- design database structure
- enable authorisation to external API
- complete user stories allocated
- monitor risk
- update risk management strategy if detect new risk
- update development speed

Sprint-2 Goal

Sprint goal	List of feature user stories
Implement school teacher booking process	<p>US-0.1 see User Stories</p> <p>As Tony Lee, I want to book a workshop through any method listed on the current website so that I can visit the place with students.</p> <p>Epic Feature 1 see User Stories</p> <p>As Emily Zhang, I want to input phone/ email bookings for clients so that I can process the booking.</p>
Implement learning team booking management	<p>Epic Feature 1 see User Stories</p> <p>As James Parker, I want to manage all pending bookings so that I can check with the booking checklist.</p>
Implement connection with Priava venue booking system	<p>Epic Feature 2 see User Stories</p> <p>As James Parker, I want to check a booking time with Priava venue booking so that I can manage the availability of the venue.</p>
Initialise database connection and storage	<p>Epic Feature 4 see User Stories</p> <p>As James Parker, I want to see all the booking information so that I can update the details.</p>

For detailed User Stories assigned to Sprint 2, See [Sprint-2 Burn Down Chart](#)

Sprint-2 Burn Down Chart



Task In Plan	Story Point (Total/Finished)
US 0.1	1
US 1.1	3 / 1.8
US 1.2	3 / 3
US 1.4	3 / 1.5
US 1.5	3 / 1.5
US 2.1	5 / 5
US 2.2	3 / 3
US 4.1	3 / 1
Total Story Point	23 / 16.8

Other User Story Finished	Story Point (Total/Finished)
US 0.2	3 / 3
US 1.3	3 / 3

Estimate Velocity= 23

Actual Velocity=22.8

US 1.1, 1.5, 4.1 are still under development as the team faces some technical issues, these user stories will be completed in the next sprint.

Task	Date	Ideal	Actual
Start	Mar 25	23	23
	Mar 30	20	20
	Apr 4	18	16
	Apr 9	15	14
	Apr 14	12	13
	Apr 19	9	10
	Apr 24	6	7
	Apr 27	3	4
End	May 1	0	0.2

Sprint-2 Review

We will arrange meetings next week with client around 1st - 7th May

3 SPRINT-3

Sprint-3 Plan

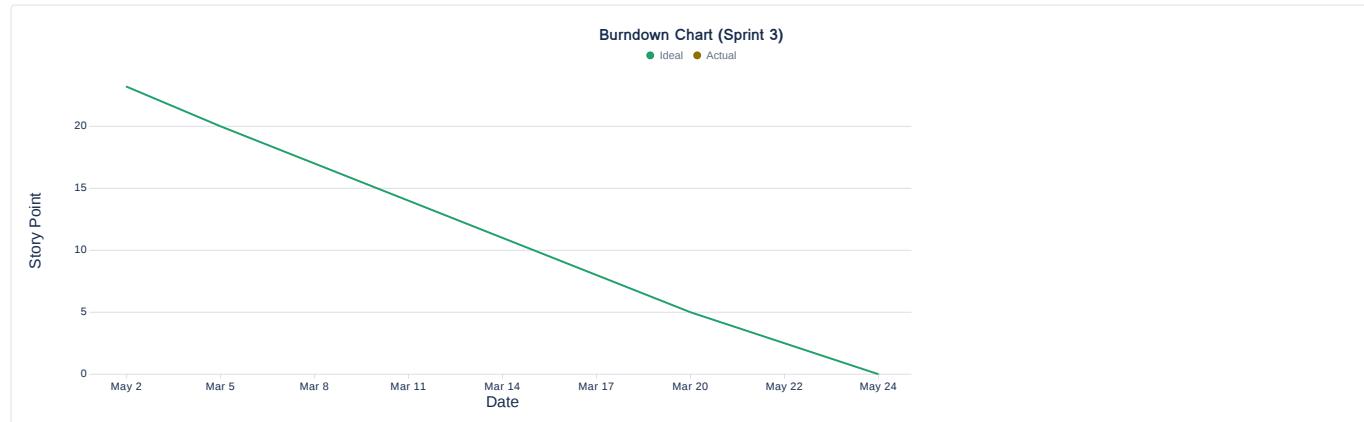
- client review on sprint 2
- complete unfinished user stories in [sprint 2](#)
- update requirements based on client review
- continue development on allocated user stories
- update risk management strategy if detect new risk
- test and document completed features
- complete user stories allocated
- update development speed

Sprint-3 Goal

Sprint goal	List of feature user stories
Finish Epic feature 0	<p>Epic Feature 0, See User Stories</p> <p>As Tony Lee, I want to book a workshop and get a booking ID so that I can update/cancel any booking later through email or call using this booking ID.</p>
Implement Microsoft Calendar management system	<p>Epic Feature 3, See User Stories</p> <p>As James Parker, I want to update booking information to Microsoft Calendar so that I can invite and roster facilitators.</p>
Enable updated booking details	<p>Epic Feature 1, See User Stories</p> <p>As James Parker, I want to update the current booking detail so that I can have the up to date information for reporting.</p>
Implement accessibility arrangements including interpreter or transportation	<p>Epic Feature 5, See User Stories</p> <p>As James Parker, I want to book additional arrangements so that I can satisfy clients' requirements.</p>
Implement update for risk assessment	<p>Epic Feature 6, See User Stories</p> <p>As James Parker, I want to send and update workshop risk assessment for a school so that they can prepare for this visit and I can keep a record of the risk of this event.</p>
Implement updates for medical information	<p>Epic Feature 7, See User Stories</p> <p>As James Parker, I want to gather and manage medical information, allergies, or accessibility needs for the visiting groups so that I can ensure a safe and accommodating visit for all attendees.</p>

For detailed User Stories assigned to Sprint 2, See [Sprint-3 Burn Down Chart](#)

Sprint-3 Burn Down Chart



Task In Plan	Story Point (Total/Finished)
US 0.2	3
US 3.1	1
US 3.2	3
US 4.3	1
US 5.1	1
US 5.2	3
US 6.1	1
US 6.2	1
US 7.1	3
Task Left from Sprint 2	
US 1.1	1.2
US 1.4	1.5
US 1.5	1.5
US 4.1	2
Total Story Point	23.2

Estimate Velocity= 22.8

Actual Velocity=

Task	Date	Ideal	Actual
Start	May 2	23.2	
	Mar 5	20	
	Mar 8	17	
	Mar 11	14	
	Mar 14	11	
	Mar 17	8	
	Mar 20	5	
	May 22	2.5	
End	May 24	0	

Sprint-3 Review

4

SPRINT-4

Sprint-4 Plan

Sprint-4 Goal

Sprint-4 Burn Down Chart

Sprint-4 Review



REQUIREMENTS

1. Goal Model:

- Outlines the **main objectives and goals** of the project or product.

2. Personas:

- Represents **fictional characters** created based on the research to represent the **different user types** who might use the product.

3. Prototypes:

- **Early models** or mockups of the product to explore ideas, design concepts, and usability testing.

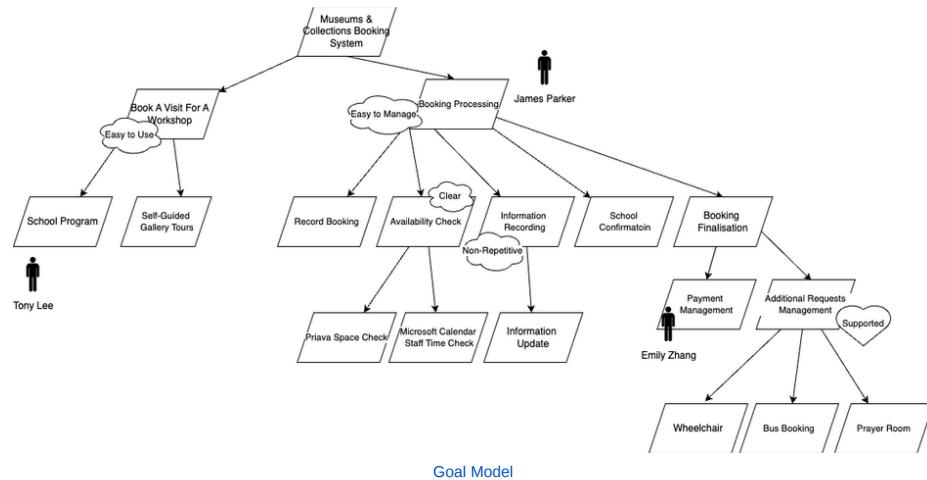
4. User Stories - Product Backlog:

- User stories describe the **features** and **functionality** of the product from the **end user's perspective**.
- The **product backlog** is a **prioritized list of user stories** that provide a comprehensive view of all the work needed for the project.

5. Provided Information - user stories support details:

- Includes **additional details** and information that support the user stories.

Goal Model



Personas



Image:generate By AI

James Parker
Booking manager

Age : 38
Gender : Male
Address : Melbourne

.....

Major Responsibilities

- Managing and overseeing the booking system for various trips within the Science Gallery.
- Ensuring a smooth and efficient booking process for both individual visitors and group tours.
- Coordinating with other departments to update trip schedules, availability, and special events.
- Handling customer inquiries and issues related to bookings.

Physical, Social, and Technological Environment

- Works in an office within the Science Gallery but often moves around the gallery to coordinate with different departments.
- Interacts with a diverse team, including exhibitions coordinators, teacher, and school representatives.
- Uses a variety of technology solutions, including a booking management system, Phava system.

Goals and Tasks

- To simplify the booking process for all users and make sure that it is user-friendly and accessible for all the user.
- To increase the efficiency of managing bookings, including ease of updating trip schedules and availability.
- To improve customer satisfaction.

Quote

"Our main duty is to let the client can book the journey of discovery efficiently and easily. To achieve this goal, my mission is to make the process as seamless as possible, ensuring that every guest are happy with the whole process."

SKILLS

Managing	★★★★★
Problem Solving	★★★★★
Teamwork	★★★★★

PERSONALITY

85%	Personal Thinking
75%	Open Minded



Image:generate By AI

Tony Lee
School Teacher

Age : 38
Gender : Male
Address : Melbourne

.....

Major Responsibilities

- Organize educational outings for students, such as visits to the Science Museum.
- Book a trip to the Science Museum for students of all ages.
- Coordinate with the Science Museum to ensure the itinerary is both educational and fun.
- Manage all trip-related logistics and safety measures.

Physical, Social, and Technological Environment

- Work within the school and communicate frequently with teachers, parents and students.
- Use your school's educational technology resources and external booking platforms to organize your trip.

Goals and Tasks

- Find and book educational Science Gallery tours that inspire students' curiosity and passion for learning.
- Make sure the booking process is quick and easy so that more time can be spent preparing educational content rather than administrative work.

Quote

Our goal is to create a lively learning experience for students through the exploration and understanding of different knowledge.

SKILLS

Teaching	★★★★★
Problem Solving	★★★★★
Teamwork	★★★★★

PERSONALITY

85%	Personal Thinking
75%	Open Minded



Image:generate By AI

Major Responsibilities

- Answering phone calls from schools and educators interested in booking educational programs.
- Processing booking requests, including entering data into the booking system, scheduling, and confirming bookings.
- Coordinating with other departments to ensure availability and readiness of programs.

Physical, Social, and Technological Environment

- Works in a busy office environment, part of the larger educational outreach department.
- Regularly interacts with educators, school administrators, and internal staff from various departments.
- Uses a specialized booking system, email, and phone as primary tools for communication and booking management.

Goals and Tasks

- To efficiently process all booking requests, ensuring schools receive prompt responses and high satisfaction with the booking experience.
- To manage the booking calendar effectively, maximizing the utilization of available programs and resources.

Quote

"Every call is an opportunity to make a positive impact on students' learning experiences. My goal is to ensure every school finds the perfect program for their needs."

Emily Zhang
Booking Specialist

Age : 28
Gender : Female
Address : Melbourne

.....

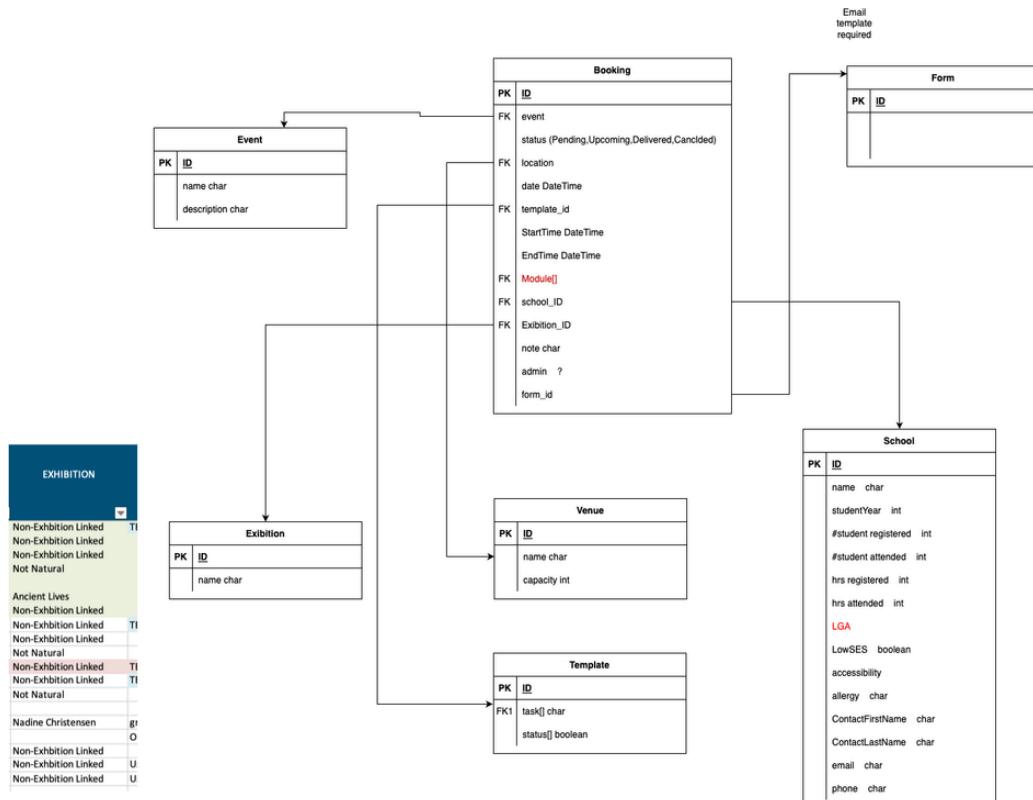
SKILLS	PERSONALITY							
Time management	★★★★★	85%						
Problem Solving	★★★★★	75%						
Teamwork	★★★★★	Humor						

Prototypes

Figma

View our design via [Booking Management System](#)

Database ER Diagram Prototype



User Stories - Product Backlog

The following user stories are based on the requirements given by the client.

Acceptance criteria is confirmed with client

- **Size Estimation/Story Points** estimate using the Fibonacci sequence, where 1 is 12 hrs, 3 is 48 hrs, 5 is half a week, 8 is a week
- **Priority** - must have, should have, could have, will not have
- **As** - reflect to [Personas](#)

Story Points	Estimated Working Hours
1	12 hours
3	48 hours
5	84 hours / half a week
8	168 hours / a week

Priority	Justification
must have	The user story is needed to satisfy client requirements and is included in Inner Scope
should have	The user story is an ideal way to satisfy client requirements and not necessarily be included in Inner Scope
could have	The user story is not essential to satisfy client requirements and is not included in Inner Scope . It is beneficial to have to improve user experience
will not have	The user story is not needed to satisfy client requirements and is not included in Inner Scope

Epic Feature 0: School teacher booking								
ID	As	I want to	So that	Acceptance Criteria	Priority	Size Estimation	Dependency	Justification
0.1	Tony Lee Client suggest that he wants to manually copy and past the booking details in the booking system. In this case 0.1 has been removed	fill in a booking form	I can book a visit for a workshop	<ul style="list-style-type: none"> • Display booking form • Must have required field of input • Let user see booking completion notification 	Must have	±	N/A	Size estimation: The current team relies on Microsoft booking form. We can adjust it to fit it in our user stories. Priority estimation: We need a booking form before

								processing the booking information by learning team member.
0.2	Tony Lee	be able to see a booking ID in my email inbox	I can update/cancel any booking later through email or call using this booking ID	<ul style="list-style-type: none"> Send email to client inbox Display booking ID 	Should have	3	0.1	<p>Size estimation: We will need to extract client email and generate unique booking ID.</p> <p>Priority estimation: Is good to have a booking ID for client. This allows the learning team staff to track booking easily by referencing the booking ID when update/cancel booking.</p>

Epic Feature 1: Learning team member booking checklist								
ID	As	I want to	So that	Acceptance Criteria	Priority	Size Estimation	Dependency	Justification
1.1	Emily Zhang	manually enter or copy the booking information if client booked by telephone/email	I can keep track of the client details and process booking.	<ul style="list-style-type: none"> Display form input field Allow user to tag the booking (based on checklist template type) Allow user to see completion notification A text field for user to paste the whole message including Program, Date, 	Must have	3	N/A	<p>Size estimation: A simple form for input and creating a record in the database</p> <p>Priority estimation: It is important that we have this method so the staff can process these booking made by phone call or email.</p>

				School, Time... <ul style="list-style-type: none">• Extract the client details from specific structure				
1.2	James Parker	tag all pending booking by checklist template type	I can keep track of all bookings process based on the template checklist	<ul style="list-style-type: none">• Display a list of pending bookings• Allow user to distinguish between different tags	Must have	3	N/A	<p>Size estimation: Only require to use distinguishable tags such as different colour and reflect on the different template</p> <p>Priority estimation: Need this for going through checklist for different types of workshop booking</p>
1.3	James Parker	update checklist template	I can have a more up to date checklist template for tracking booking progress	<ul style="list-style-type: none">• Allow user to see current template by name• Allow user to only update the required field	Could have	3	N/A	<p>Size estimation: Update all booking checklist with the same tagged template, need to search and update all relevant bookings in database.</p> <p>Priority estimation: not the most important, but it will be flexible to have it.</p>
1.4	James Parker	see all bookings and see each booking checklist progress	I can track progress of all bookings and individual one's	<ul style="list-style-type: none">• Allow user to see all booking details• Allow user to search for target booking	Should have	3	1.1	<p>Size estimation: Need searching for one individual booking for convenience.</p> <p>Priority estimation: Learning team member should be seeing this if they want to track the</p>

									progress using the checklist
1.5	James Parker	update booking details	I can track the more up to date booking information.	<ul style="list-style-type: none"> Allow user to see all booking details Allow user to search for target booking Allow user to only update the required field 	Should have	3		1.1, 1.4	<p>Size estimation: Need searching for target booking for convenience. Need update target booking details in all other platform</p> <p>Priority estimation: This is a necessary functionality, but automation can be develop later in the development.</p>

Epic Feature 2: Connection and negotiation with Priava venue booking system								
ID	As	I want to	So that	Acceptance Criteria	Priority	Size Estimation	Dependency	Justification
2.1	James Parker	check Priava venue availability	I know if there is space available for a booking	<ul style="list-style-type: none"> Allow user to login to Priava Allow user to see Priava information inside the booking system 	Must have	5	N/A	<p>Size estimation: Need to use Priava API to embed in our booking system, setup Authentication to use this API</p> <p>Priority estimation: Priava is essential in this booking system</p>
2.2	James Parker	have booking details copied into Priava if there is availability	I can have the venue booked	<ul style="list-style-type: none"> Allow user to submit booking details to priava 	Must have	3	1.1, 1.2, 2.1	<p>Size estimation: Need to linked to Priava API for automation</p> <p>Priority estimation: This is required for venue booking and booking information need to be shown in Priava as well.</p>

Epic Feature 3: Microsoft Calendar management								
ID	As	I want to	So that	Acceptance Criteria	Priority	Size Estimation	Dependency	Justification
3.1	James Parker	add Priava booking information to the shared Microsoft Calendar	we can provide facilitators with immediate access to venue availability	<ul style="list-style-type: none"> Display correct time slot Information should be same as one given from Priava 	Must have	1	2.1,2.2	<p>Size estimation: Need to linked to Microsoft API for automation</p> <p>Priority estimation: This is required for venue booking and booking information need to be shown in Microsoft Calendar</p>
3.2	James Parker	invite facilitators to assist with scheduling	we can ensure the schedules are processed and audited efficiently	<ul style="list-style-type: none"> if casual facilitator required, roster via When I Work 	Must have	3	3.1	<p>Size estimation: Need to associate with When I work</p> <p>Priority estimation: facilitator is essential process and audit the tasks</p>

Epic Feature 4: Store booking details in Database								
ID	As	I want to	So that	Acceptance Criteria	Priority	Size Estimation	Dependency	Justification
4.1	James Parker	see a list of booking details in the booking system	I can select a booking and update the booking information	<ul style="list-style-type: none"> Display grouped data Allow user to update booking details Allow user to search a booking by booking id 	Must have	3	1.1, 1.2	<p>Size estimation: Link database to the booking system. Data should be stored once booking details was entered by client or learning team members.</p> <p>Priority estimation: Required for learning team member to track historical bookings.</p>
4.2	James Parker	see analysis of the booking details	I can report to the team	<ul style="list-style-type: none"> Display visual analysis 	Could have	3	1.1, 1.2, 4.1	<p>Size estimation: Need to process booking data and need to use</p>

				<ul style="list-style-type: none"> Display data trend 				packages for analysis.
4.3	James Parker	update a booking information	I can have the up to date information for reporting	<ul style="list-style-type: none"> Display most recent booking details Allow user to see update success notification 	Should have	1	4.1	<p>Size estimation: Need to update data in the database and show pop out video for update success notification.</p> <p>Priority estimation: Not necessarily needed in the booking system, but good to have for analytic task.</p>

Epic Feature 5: Accessibility Arrangements for booking								
ID	As	I want to	So that	Acceptance Criteria	Priority	Size Estimation	Dependency	Justification
5.1	James Parker	easily book interpreters through Auslan Services	deaf or hard of hearing students can fully engage with the visit.	<ul style="list-style-type: none"> Interface to select the need for an Auslan interpreter during booking 	Could have	1	3.1	<p>Size estimation: System should able to connect to Auslan Services including the process for requesting and confirming interpretation services.</p> <p>Priority estimation: Ensuring accessibility is very important for inclusive</p>

								educational experiences.
5.2	James Parker	offer and book transportation for schools requiring a bus to our venue through BusCharter.com.au	logistical challenges do not prevent schools from attending.	<ul style="list-style-type: none"> Option to request bus transport during the booking process. Interface to book buses directly or send requests to Bus Charter & Coach Hire in Australia Confirmation of bus booking sent to the school representative . 	Should have	3	3.1	<p>Size estimation: Need to use api to connect to BusCharter.com.au and handle the transport request and confirmation result.</p> <p>Priority estimation: Promote school participation and make the whole process easier.</p>

Epic Feature 6: Risk								
ID	As	I want to	So that	Acceptance Criteria	Priority	Size Estimation	Dependency	Justification
6.1	James Parker	send prewritten risk assessments to schools for all our programs	teachers can prepare and address any concerns prior to the visit.	<ul style="list-style-type: none"> Automated distribution of risk assessment s upon booking confirmation. 	Could have	1	1.1, 1.2	<p>Size estimation: Enable automatic sending of risk assessments</p> <p>Priority estimation: It is very important for schools to understand possible risks before visiting and to promote a safe environment for students.</p>
6.2	James Parker	confirm booking details with teachers two weeks	any necessary adjustments can be made in a timely manner.	<ul style="list-style-type: none"> Automated reminder to send confirmation 	Could have	1	3.1	<p>Size estimation: Enable automatic reminder and</p>

		before their scheduled visit		emails two weeks before the booking.				confirmation email sending Priority estimation: Reduces the potential for misunderstandings is very important.
--	--	------------------------------	--	--------------------------------------	--	--	--	---------------------------------------------------------------------------------------------------------------------------------

Epic Feature 7: Health and Safety Information

ID	As	I want to	So that	Acceptance Criteria	Priority	Size Estimation	Dependency	Justification
7.1	James Parker	gather and manage medical information, allergies, or accessibility needs for the visiting groups	we can ensure a safe and accommodating visit for all attendees.	<ul style="list-style-type: none"> Secure form for collecting sensitive medical and accessibility information during the booking process. 	Must have	3	3.1	Size estimation: Collecting sensitive medical and accessibility information requires keeping the data secure and private and the data can be easily accessible and manageable by internal personnel. Priority estimation: It is vital to prepare and provide a safe, inclusive visiting environment that meets the needs of all participants.

Epic Feature 8: Post-visit Reporting

ID	As	I want to	So that	Acceptance Criteria	Priority	Size Estimation	Dependency	Justification
8.1	James Parker	accurately record and report the number of students who attended each	we can evaluate our outreach and report on our educational impact.	<ul style="list-style-type: none"> Interface for entering attendance numbers into the LER database. 	Must have	3	N/A	Size estimation: Implement an interface to enter and update attendance numbers in the database

	program in our database				Priority estimation: Providing valuable data to assess program effectiveness and plan future delivery is a must.
--	-------------------------	--	--	--	----------------------------------------------------------------------------------------------------------------------------

Epic Feature 9: Financial Processing								
ID	As	I want to	So that	Acceptance Criteria	Priority	Size Estimation	Dependency	Justification
9.1	James Parker	process payments for interpreters and buses through the university's financial system	external providers are paid promptly and the service is secured for the schools.	<ul style="list-style-type: none"> Integration with the university's financial system for direct payment processing. Automated payment confirmation to both the service providers and school representatives. 	Must have	3	3.1	<p>Size estimation: Requires integration with the University's financial systems and automation of payment processing.</p> <p>Priority estimation: Prompt payment to school is key to maintaining a good relationship.</p>

Provided Information - user stories support details

Topic		
Planner	The staff member copies the booking details into Microsoft Planner/Tasks into the checklist template so they can track the steps of the booking process.	
Priava	The staff member checks Priava (venue booking system) that the workshop space or gallery space is available to be used, then copies the booking information into a Priava booking. If there is not sufficient space, they will negotiate with the school around an alternative date and time	
MSC + Invites	Once the Priava booking is confirmed, the staff member will add the booking to a shared Microsoft Outlook calendar, once again copying over the booking information. Outlook is used as the main tool by our team as it allows us to invite facilitators to the programs to assist with their own scheduling. If casual facilitators are required, they will also be rostered on here using When I Work.	
LER + update status	This is the stage where the staff member will enter the data into our excel spreadsheet database of all the programs we have run, manually entering all the data that we capture from the booking process. We also use this data for our reporting. For example, this would be (how many students from X school attended in term 1, and how many total hours did they spend here).	
Additional resources - booking is also recorded in database link to the target workshop booking	<p><i>Book interpreters</i></p> <p>Some of our schools have deaf/hard of hearing students that require interpreters. These are booked through Auslan Services.</p> <p><i>Book bus if needed</i></p> <p>If a partner school would like to have a bus to our venues, we will book a bus through Bus Hire - Low Cost Bus Charter & Coach Hire in Australia.</p> <p><i>Paid for interpreter/bus</i></p> <p>Our staff member will process the payments for buses and interpreters through the university system. This is currently done through Themis but the university is changing this system some time in the year.</p>	
Update information in database	<p><i>Risk assessments</i></p> <p>We have prewritten risk assessments for all our programs that we send out to schools.</p>	

<i>Confirm with teacher</i>	the staff member will send out a confirmation email two weeks before the booking to make sure the details are still correct.
<i>Medical information</i>	the staff member will obtain any medical information, allergies or accessibility needs for the group and update the Outlook calendar with the information.
<i>Numbers on the day</i>	We need to enter how many students attended into our database spreadsheet (the LER). This is for reporting purposes.

 RISK1. **Risk Registry:**

- Includes information such as the **description of the risk, its impact, likelihood, mitigation strategies**, and responsible person for managing the risk.
- The registry is regularly updated to reflect new risks, changes in risk status, and the effectiveness of risk response strategies.

2. **Review Meetings:**

- Scheduled meetings to **discuss and review the current risk status**, effectiveness of mitigation actions, and identification of new risks.

Risk Registry

Risk ID	Description	Category	Probability	Impact	Risk Level	Mitigation Plan	Owner	
0	API failure leading to interruption in service delivery	Technical	Medium	High	High	Implement alternative APIs; Regular API health checks; Develop a rapid response plan for API failure	@Guixian Li	
1	Software updates to the platform or database causing incompatibility issues	Technical	Medium	High	High	Test any software updates in a controlled environment before rolling them out	@Yuanbo Xu	
2	Data leakage in database due to unsecured data	Security	Low	High	Medium	Preprocess data before saving to ensure no sensitive information is stored	@Yuanbo Xu	[Date]
3	Inaccurate project time estimation leading to delays	Management	High	Medium	High	Implement more rigorous time estimation techniques	@Guixian Li	

Review Meetings



COLLABORATION

1. About Us:

- Provides an **overview of the team**.

2. Team Calendar:

- A shared calendar that **tracks all team-related events, meetings, deadlines**, and important milestones.

3. Agile Processes:

- Describes the **specific Agile methodologies and practices** adopted by the team.

4. Agile Ceremonies:

- Refers to the **essential meetings or events** in Agile methodologies that help teams to implement Agile practices effectively.
- Includes ceremonies like **Sprint Planning, Sprint Review, and Sprint Retrospective**.

About Us

Team Info

NAME	ROLE	CONTACT
@Yujun Yan	Mentor	?
@Guixian Li <i>Leona</i>	Product Owner	guixianl@student.unimelb.edu.au
@Daniel Su <i>Daniel</i>	Developer	dss1@student.unimelb.edu.au
@Yuanbo Xu <i>Chris</i>	Developer	yuanbo@student.unimelb.edu.au
@Pangfeng ZHENG <i>Jack</i>	Scrum Master	pangfenz@student.unimelb.edu.au
@Yongli Qin <i>Matt</i>	Developer	yongliq@student.unimelb.edu.au
@Yun-Chi Hsiao <i>Jim</i>	Recorder	yunchi@student.unimelb.edu.au

External

- GitHub <https://github.com/COMP90082-2024-SM1/SG-Koala> Connect your Github account
- Trello [COMP90082_2024_SM1_SG_KOALA](#)
- Slack Channel
 - #comp90082-2024-sg-koala
 - #client-meeting
 - #general
 - #meeting

Team Calendar

Meeting Type	Meeting time	People Involved
Sprint Plan Meeting	Before each Sprint start	All Team Member
Product Backlog Refinement Meeting	Before each Sprint start	All Team Member
Weekly Stand-up Meeting	Twice a week (Monday, Wednesday)	All Team Member
Supervisor Meeting	Weekly (Wednesday)	All Team Member, Supervisor
Review Meeting	After a Sprint finished	All Team Member, Client
Retrospective Meeting	After a Sprint finished	All Team Member

Agile Processes

Team Charter

Team Purpose and Mission

- Aim to significantly reduce workload and enhance user experience.
- Develop a user-friendly, efficient software application to improve the Science Gallery's booking system.

Scope of Work

- Design, develop, and test the application.
- Interface with the Science Gallery's existing systems and databases.
- Ensure accessibility and ease of use for various user groups.

Roles and Responsibilities

- Project Manager: Manages project schedules and liaises with the Science Gallery.
- Scrum Master: Guides and assists the Scrum team, ensuring Scrum practices are followed, resolving obstacles, and promoting a productive team environment.
- Recorder: Documents the development process comprehensively, including creating user guides.
- Developers: Responsible for coding and building the software application, including writing, testing, and debugging code, and collaborating on design and implementation decisions.

Procedures and Processes

- Weekly team meetings to discuss current work.
- Weekly meetings with Supervisor for progress reviews.
- Sprint sessions include planning, retrospectives, and client reviews.
 - Planning: Organising and defining tasks and objectives for the sprint.
 - Retrospective: Reflecting on the past sprint to identify improvements.
 - Review with Clients: Discussing progress and feedback with clients.
- Use of collaboration tools.
 - Trello for task management.
 - Slack for communication.

Resources

- Team members will independently research and document resources, compiling them in a shared document [🔗 LINKS & RESOURCES](#) for easy access and reference.
- Additionally, the team will coordinate with the client to obtain a testing account and necessary data for effective system testing and validation.

Performance Metrics

- Completion of development milestones on schedule.
- Reduction in booking system workload as reported by the Science Gallery.
- User satisfaction and feedback post-implementation.

Conflict Resolution Strategies

- Encourage open and respectful communication to address issues as they arise.
- Implement regular check-ins to identify and resolve emerging conflicts early.
- Facilitate mediation sessions with a neutral party for unresolved conflicts.

Duration and Timeframes

The team will operate over the duration of the semester, with the project structured into four distinct sprints:

- Sprint 1: 3 weeks.
- Sprint 2: 4 weeks.
- Sprint 3: 4 weeks.
- Sprint 4: 2 weeks.

The project is set to conclude by the end of the semester, aligning with these sprint durations for effective planning and execution.

Definition of Ready

The "Definition of Ready" in software development and Agile methodologies refers to a set of criteria that a user story or task must meet before it can be considered ready to be worked on in a sprint or development cycle. This concept ensures that work items are adequately prepared and understood before the team starts working on them, promoting efficiency and reducing the risk of misunderstandings or rework. Key aspects of the Definition of Ready typically include:

- **Clear Acceptance Criteria:** Specific conditions that must be met for the work to be considered complete.
- **Dependencies Resolved:** All dependencies on other tasks or external factors are identified and resolved.
- **Estimated:** The task has been sized or estimated by the team.
- **Feasible:** The team agrees that the task is possible to complete within the sprint or cycle.
- **Prioritised:** The task fits within the team's priorities and product roadmap.
- **Understood by the Team:** Everyone involved has a clear understanding of what needs to be done and why.

The Definition of Ready is a crucial component of Agile project management, ensuring that tasks are well-prepared before development begins, thus improving the team's efficiency and effectiveness.

Definition of Done

The "Definition of Done" in Agile and software development is a critical agreement that determines when a task or user story is considered complete. It serves as a checklist that ensures quality and completeness. Key aspects typically include:

- **Quality Standards Met:** The work meets predefined quality criteria, such as code quality, design standards, and performance benchmarks.
- **Acceptance Criteria Fulfilled:** The task satisfies all the conditions specified in the user story or requirement.
- **Code Review and Testing:** The code has been reviewed, and all necessary testing (unit tests, integration tests, etc.) has been successfully completed.
- **Documentation Updated:** Relevant documentation (technical, user manuals, etc.) has been updated to reflect the changes.
- **Approved by Stakeholders:** The work is reviewed and approved by the relevant stakeholders, such as product owners, clients, or team leads.
- **Potentially Shippable:** The increment is in a state where it could be released to users, even if it won't be shipped immediately.

The Definition of Done ensures transparency and a shared understanding of what it means for work to be complete, helping to avoid ambiguities and ensuring that each product increment is of high quality.

Agile Ceremonies

Sprint Planning - 1

Date & Time

13:00 Mar. 13th, 2024

Participants

- @Yuanbo Xu
- @Pangfeng ZHENG
- @Yongli Qin
- @Daniel Su
- @Yun-Chi Hsiao
- @Guixian Li

Goals

- Determine the specific roles for each team member.
- Allocate specific tasks to each team member.

Discussion Topics

Time	Item	Presenter	Notes
10 mins	Roles	@Guixian Li	Discuss each team member's personality and capabilities to assign roles effectively.
30 mins	Tasks	@Guixian Li	Compile the tasks for Sprint 1 using the checklist provided on the LMS.

Action Items

- Determine the specific roles for each team member.
- Allocate specific tasks to each team member.

Decisions

👉 @Guixian Li will be the **project owner**.

👉 @Yuanbo Xu is the **scrum master**.

👉 @Yun-Chi Hsiao is the **recorder**.

👉 @Guixian Li @Yuanbo Xu @Yongli Qin are responsible for **REQUIREMENTS** section.

👉 @Pangfeng ZHENG takes responsibility for overseeing **Trello** and planning **SPRINT-1**.

👉 @Yun-Chi Hsiao manages **Confluence**, **GitHub**, and **COLLABORATION** content.

 @Daniel Su is responsible for the  BACKGROUND section.

Retrospective - 1

Date & Time

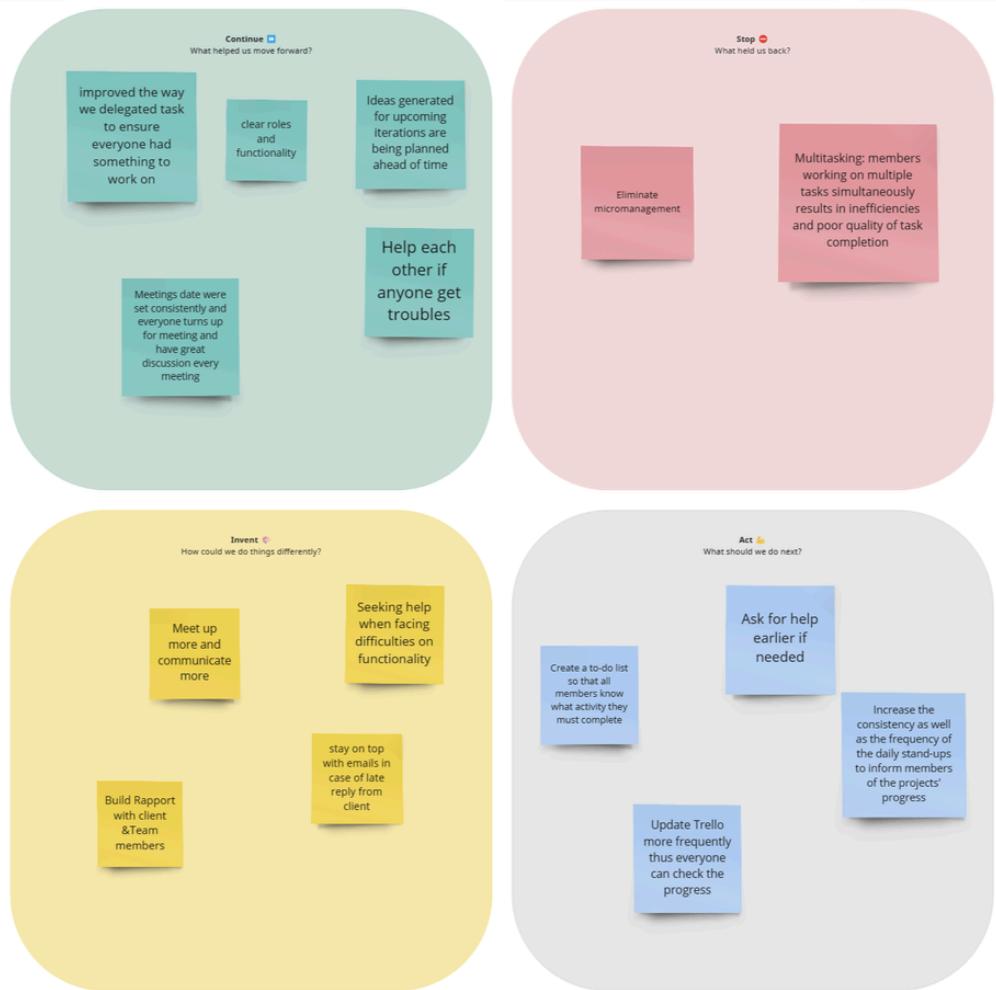
11:00 Mar. 27th, 2024

Participants

- [@Yuanbo Xu](#)
- [@Pangfeng ZHENG](#)
- [@Yongli Qin](#)
- [@Daniel Su](#)
- [@Yun-Chi Hsiao](#)
- [@Guixian Li](#)

Goals

- Reflection and evaluation: Review all aspects of the last Sprint together, including problems encountered when performing tasks, team collaboration, and achieving goals.
- Identify successes and challenges: Identify which practices are successful and should be maintained and which challenges or problems require improvement.



Review Meeting - 1

📅 Date & Time

09:30 Mar. 27th, 2024

👥 Participants

- @Yuanbo Xu
- @Pangfeng ZHENG
- @Yongli Qin
- @Daniel Su
- @Yun-Chi Hsiao
- @Guixian Li
- The client

📋 Goals

- Review current design.
- Questions for the client.

🗣 Discussion Topics

Time	Item	Presenter	Notes
30 mins	Figma Review	The client	<ol style="list-style-type: none">1. Overall, it's well-designed.2. The search bar should enable users to find both school and non-school partners.3. A textbox for staff to paste booking details from emails should be included.
20 mins	Discussion	@Guixian Li @Yuanbo Xu @Yongli Qin @Daniel Su @Pangfeng ZHENG @Yun-Chi Hsiao	<ol style="list-style-type: none">1. Chris will supply the email template.2. When clicking a hyperlink, it will open in a new page rather than being embedded in the current page.3. Consideration of whether different clients should use the same account.

✓ Action Items

- ✓ Let's review our Figma designs and address the questions that arose during our design phase.
- ✓ We'll undertake a review of the Figma designs to discuss any issues we encountered in the design process.
- ✓ Update Figma and User Stories according to client needs. @Yongli Qin @Yun-Chi Hsiao @Guixian Li
- ✓ Improve Confluence structure and improve document quality @Yuanbo Xu @Daniel Su @Pangfeng ZHENG
- ✓ @Pangfeng ZHENG Update Trello according to user stories

Decisions

- 👉 The same account can be accessed by multiple staff members on various devices.
- 👉 Priava, Microsoft, and the bus booking system all incorporate hyperlinks.
- 👉 Incorporate a textbox for entering an email address into the form creation process.
- 👉 **Chris says we will get H1!!!**

Sprint Planning - 2

Date & Time

13:00 Mar. 27th, 2024

Participants

- @Yuanbo Xu
- @Pangfeng ZHENG
- @Yongli Qin
- @Daniel Su
- @Yun-Chi Hsiao
- @Guixian Li

Goals

- Determine the specific tasks for each team member in Sprint-2.
- Confirm user stories and design with the client
- Complete user stories assigned to Sprint-2
- Monitor potential risk

Discussion Topics

Time	Item	Presenter	Notes
10 mins	Roles, Idea	@Guixian Li	Discuss each team member's personality and capabilities to assign roles effectively. Discuss each team member's idea on Figma
30 mins	Tasks	@Guixian Li	Compile the tasks for Sprint 1 using the checklist provided on the LMS.

Action Items

- Determine the specific roles for each team member.
- Allocate specific tasks to each team member.

Decisions

👉 We split features between members based on Trello - see [Trello](#)

👉 Update prototype design based on client feedback - add a new text field in Booking form allowing clients to paste booking details and auto matching into our booking form

👉 Confirm typography

Sprint Planning - 3

Date & Time

13:00 April. 10th, 2024

Participants

- @Yuanbo Xu
- @Pangfeng ZHENG
- @Yongli Qin
- @Daniel Su
- @Yun-Chi Hsiao
- @Guixian Li

Goals

- Determine the specific tasks for each team member in Sprint-3.
- Review code from Sprint 2
- Update user stories if necessary, reestimate story points
- Update burn-down chart if necessary
- Update design prototypes - Figma
- Complete user stories assigned to Sprint-3
- Monitor potential risk

Discussion Topics

Time	Item	Presenter	Notes
10 mins	Reflect Sprint 2 Task	@Guixian Li	Discuss unfinished tasks from Sprint 2. Reflect on Sprint 2 code, design and feedback. Compile the tasks for Sprint 2 using the checklist provided on the LMS.
30 mins	Sprint 3 Tasks	@Guixian Li	Assign tasks in Sprint 3 according to each member's role and responsibility.

✓ Action Items

- ✓ Determine the specific tasks for Sprint 3.
- ✓ @Yuanbo Xu @Guixian Li Backend API, Unit testing
- ✓ @Pangfeng ZHENG Interaction between Front and Back end
- ✓ @Yongli Qin @Daniel Su @Yun-Chi Hsiao Frontend Development
- ✓ Allocate specific tasks to each team member.

Decisions

-  We split features between members based on Trello - see [Trello](#)
-  Update Prototypes to improve user experiences - see [Prototype](#)
-  Confirm Product from Sprint 2 with the client

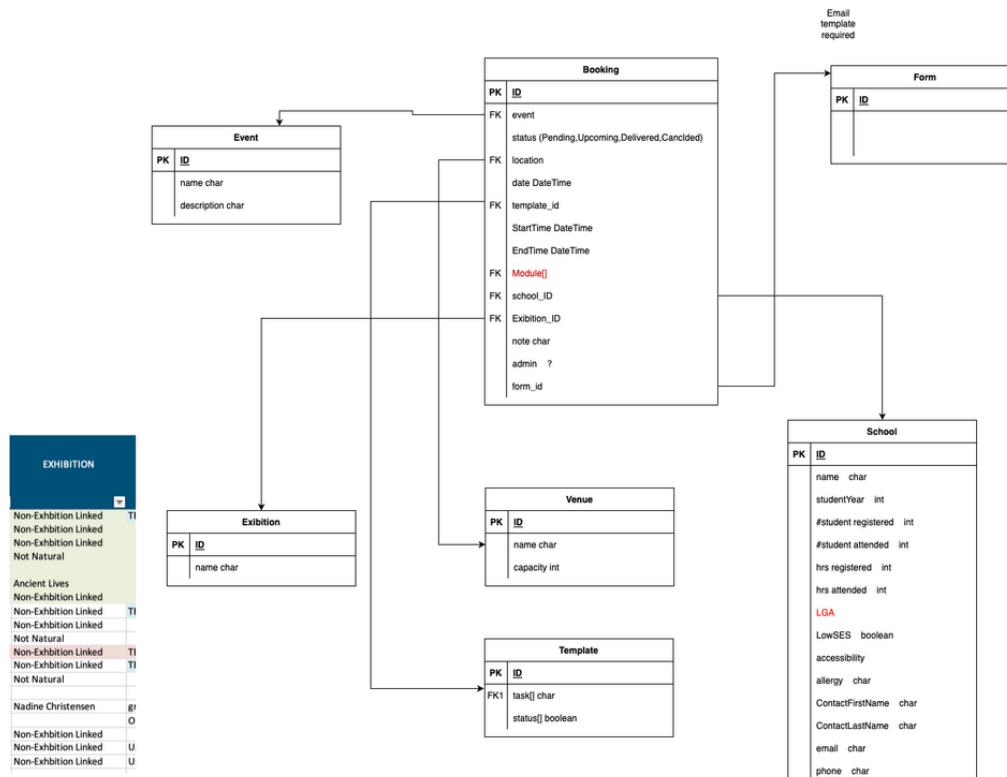
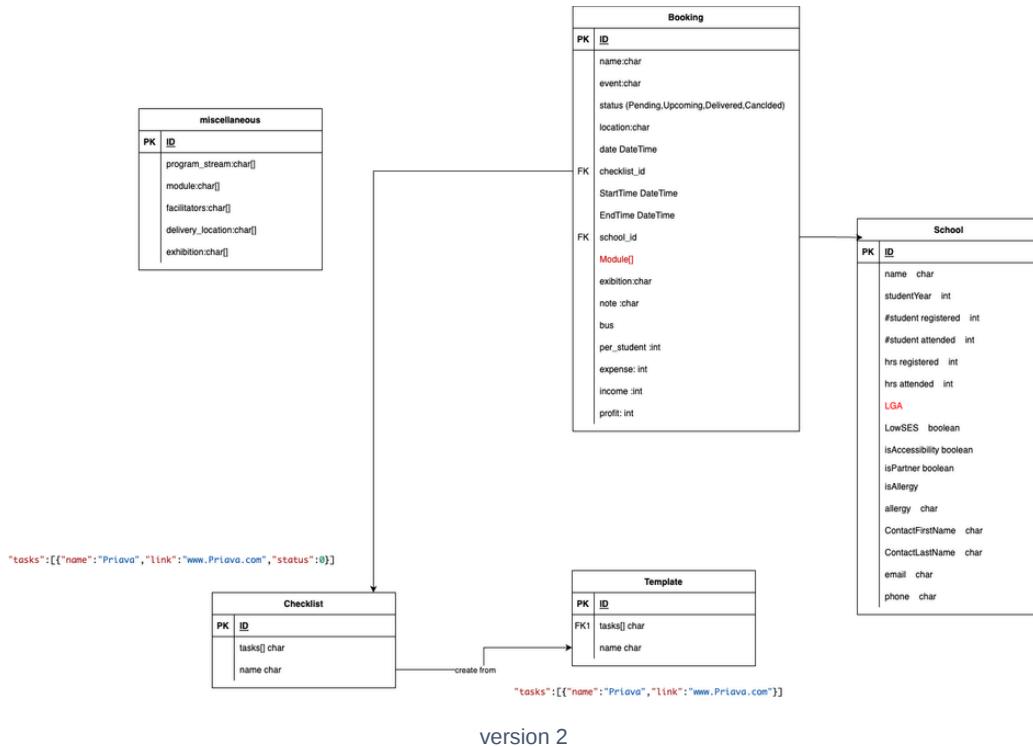
 ARCHITECTURE1. **Database Model:**

- Represents the **structure of the database** including the relationships and constraints between data entities.

2. **API (Application Programming Interface):**

- Defines the **interactions** between multiple software intermediaries, specifying how they should **communicate** and **share data**.

Database Model



API

✓ API Naming Convention Example

- GET api/ticket/ - Retrieves all tickets
- GET api/ticket/11321312adfdas/ - Retrieves a specific ticket
- POST api/ticket/ - Creates a new ticket
- PUT api/ticket/12312321fdasfsda/ - Updates ticket #12
- DELETE api/ticket/12312321fdsafdsf/ - Deletes ticket #12
- All requests end with /

📘 Postman API

<https://www.postman.com/kaibai/workspace/sc-api>



📘 Template

Method	URL Example	Request Body	Response	Description
GET	api/template/		<pre>1 [2 { 3 "id": "6614cea3647dc4b8a12f2625", 4 "name": "template1", 5 "task": [6 { 7 "name": "step1", 8 "link": "http://www.google." 9 }, 10 { 11 "name": "step2", 12 "link": "http://www.example" 13 }, 14 { 15 "name": "step3", 16 "link": "" 17 }, 18 { 19 "name": "step4", 20 "link": "" 21 } 22] 23 }, 24 { 25 "id": "6614cea3647dc4b8a12f2625", 26 "name": "template1", 27 "task": [</pre>	

```

28     {
29         "name": "step1",
30         "link": "http://www.google.com"
31     },
32     {
33         "name": "step2",
34         "link": "http://www.example.com"
35     },
36     {
37         "name": "step3",
38         "link": ""
39     },
40     {
41         "name": "step4",
42         "link": ""
43     }
44 ]
45 }
46 ]

```

GET	api/template/6614cea3647dc4b8a12f2625/	<pre> 1 { 2 "id": "6614cea3647dc4b8a12f2625", 3 "name": "template1", 4 "task": [5 { 6 "name": "step1", 7 "link": "http://www.google.com" 8 }, 9 { 10 "name": "step2", 11 "link": "http://www.example.com" 12 }, 13 { 14 "name": "step3", 15 "link": "" 16 }, 17 { 18 "name": "step4", 19 "link": "" 20 } 21] 22 } </pre>
-----	----------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

POST	api/template/	<pre> 1 { 2 "id": "6614cea3647dc4b8a12f2625", 3 "name": "template1", 4 "task": [5 { 6 "name": "step1", 7 "link": "http://www.google.com" 8 }, 9 { 10 "name": "step2", 11 "link": "http://www.example.com" 12 }, 13 { 14 "name": "step3", 15 "link": "" 16 }, 17 { 18 "name": "step4", 19 "link": "" 20 } 21] 22 } </pre>
------	---------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

		<pre> 17 { 18 "name": "step4", 19 "link": "" 20 } 21] 22 }</pre>	<pre> 17 "name": "step4", 18 "link": "" 19 } 20], 21 "_id": "661e4be0e977d7711c301ae9" 22 }</pre>	
PUT	api/template/661e4be0e977d7711c301ae9/	<pre> 1 { 2 "name": "template2", 3 "task": [4 { 5 "name": "step1", 6 "link": "" 7 }, 8 { 9 "name": "step2", 10 "link": "" 11 }, 12 { 13 "name": "step3", 14 "link": "" 15 }, 16 { 17 "name": "step4", 18 "link": "" 19 } 20] 21 }</pre>	<pre> 1 { 2 "status": "success", 3 "id": "661e4be0e977d7711c301ae9", 4 "updated": 1 5 }</pre>	
DELETE	api/template/66177abaa93b4dbfb5e70fa/		<pre> 1 { 2 "status": "success", 3 "id": "66177abaa93b4dbfb5e70fa" 4 }</pre>	

Checklist

Method	URL Example	Request Body	Response	Description
GET	api/checklist/		<pre> 1 [2 { 3 "id": "661e4c8de977d7711c301aee", 4 "name": "template2", 5 "task": [6 { 7 "name": "step1", 8 "link": "http://www.baidu.com", 9 "status": 0 10 }, 11 { 12 "name": "step2", 13 "link": "http://www.example.com", 14 "status": 0 15 }, 16 { 17 "name": "step3", 18 "link": "http://www.test.com", 19 "status": 0 20 } 21] 22 }</pre>	

```

18         "link": "",
19         "status": 0
20     },
21     {
22         "name": "step4",
23         "link": "",
24         "status": 0
25     }
26   ]
27 }
28 ]

```

GET	api/checklist/66 1e4c8de977d7 711c301aee/	<pre> 1 { 2 "id": "661e4c8de977d7711c301aee", 3 "name": "template2", 4 "task": [5 { 6 "name": "step1", 7 "link": "http://www.baidu.com", 8 "status": 0 9 }, 10 { 11 "name": "step2", 12 "link": "http://www.example.com", 13 "status": 0 14 }, 15 { 16 "name": "step3", 17 "link": "", 18 "status": 0 19 }, 20 { 21 "name": "step4", 22 "link": "", 23 "status": 0 24 } 25] 26 } </pre>
-----	-------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

POST	api/checklist/ 66 1e4be0e977d7 711c301ae9/	<pre> 1 { 2 "name": "template2", 3 "task": [4 { 5 "name": "step1", 6 "link": "http://www.baidu.com", 7 "status": 0 8 }, 9 { 10 "name": "step2", 11 "link": "http://www.example.com", 12 "status": 0 13 }, 14 { 15 "name": "step3", 16 "link": "", 17 "status": 0 18 }, 19 { 20 "name": "step4", 21 "link": "", 22 "status": 0 23 } 24] 25 } </pre>	To create a checklist, you need use a valid template ID
------	-----------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------

			<pre> 21 "link": "", 22 "status": 0 23 } 24], 25 "_id": "661e4c8de977d7711c301aee" 26 }</pre>	
PUT	api/checklist/661e4c8de977d7711c301aee/		<pre> 1 { 2 3 "name": "temp" 4 "task": [5 { 6 "name": "link" 7 "link": "http://example.com/task1" 8 "status": 0 9 }, 10 { 11 "name": "link" 12 "link": "http://example.com/task2" 13 "status": 0 14 }, 15 { 16 "name": "link" 17 "link": "http://example.com/task3" 18 "status": 0 19 }, 20 { 21 "name": "link" 22 "link": "http://example.com/task4" 23 "status": 0 24 } 25] 26 }</pre>	<pre> 1 { 2 "status": "success", 3 "id": "661e4c8de977d7711c301aee", 4 "updated": 1 5 }</pre>
DELETE	api/checklist/6603d23b9e45d1f805c5fdd1/			<pre> 1 { 2 "status": "success", 3 "id": "6603d23b9e45d1f805c5fdd1" 4 }</pre>

Booking

Method	URL Example	Request Body	Response	Description
GET	api/booking/		<pre> 1 [2 3 { 4 "id": "661633364ee0ba663dcf23c3", 5 "name": "Annual Science Fair", 6 "event": "Science Fair 2024", 7 "status": "Processing", 8 "location": "Exhibition Center - Hall A", 9 "date": "2024-09-05T09:00:00Z", 10 "checklist_id": "661631c15ca3f59d6", 11 "startTime": "2024-09-05T10:00:00Z", 12 "endTime": "2024-09-05T16:00:00Z", 13 "duration": "06:00:00" 14 } 15] 16 }</pre>	

```

13     "school_id": "661630a9577bed19d130",
14     "exhibition": "Local Innovators Show",
15     "note": "Extra chairs needed in the hall",
16   },
17   {
18     "id": "661633751a32e0d0bd170449",
19     "name": "Annual Science Fair",
20     "event": "Science Fair 2024",
21     "status": "Processing",
22     "location": "Exhibition Center - Hall A",
23     "date": "2024-09-05T09:00:00Z",
24     "checklist_id": "661e4fdae977d7711c301af",
25     "checklist": [
26       {
27         "id": "661e4fdae977d7711c301af",
28         "name": "Paid Workshop",
29         "task": [
30           {
31             "name": "Planner (from LER & Roster)",
32             "link": "",
33             "status": 0
34           },
35           {
36             "name": "Make initial contact with exhibitors",
37             "link": "",
38             "status": 0
39           },
40           {
41             "name": "Note: Take Feedback from exhibitors",
42             "link": "",
43             "status": 0
44           },
45           {
46             "name": "If struggling to get responses, follow up",
47             "link": "",
48             "status": 0
49           },
50           {
51             "name": "Update holds on roster based on responses",
52             "link": "https://apac-schools.com/roster-updates",
53             "status": 0
54           },
55           {
56             "name": "Add to LER & Roster if not already present",
57             "link": "",
58             "status": 0
59           },
60           {
61             "name": "Respond with thanks and next steps",
62             "link": "",
63             "status": 0
64           },
65           {
66             "name": "If Future Focus, add to follow-up list",
67             "link": "",
68             "status": 0
69           },
70           {
71             "name": "KE to roster for tracking and reporting",
72             "link": "",
73             "status": 0
74           }
75         ]
76       }
77     ]
78   }
79 }
```

```

73         },
74         {
75             "name": "Check in with",
76             "link": "",
77             "status": 0
78         }
79     ],
80     {
81         "startTime": "2024-09-05T10:00:00Z",
82         "endTime": "2024-09-05T16:00:00Z",
83         "school_id": "661630a9577bed19d130",
84         "school": {
85             "name": "Greenwood Elementary",
86             "studentYear": 2020,
87             "numStudentAttended": 150,
88             "numStudentRegistered": 200,
89             "hourRegistered": 1200,
90             "hourAttended": 1100,
91             "lowSES": true,
92             "allergy": "Peanuts",
93             "contactFirstName": "Jane",
94             "contactLastName": "Doe",
95             "email": "jane.doe@greener.com",
96             "phone": "555-1234",
97             "note": "",
98             "isAccessibility": false,
99             "isAllergy": false,
100            "isPartner": false
101        },
102        "exhibition": "Local Innovators Show",
103        "note": "Extra chairs needed in the hall"
104    }
105 ]

```

GET

api/booking/6603d23b9e45d1f805c5fdd1/

```

1  {
2      "id": "661633364ee0ba663dcf23c3",
3      "name": "Annual Science Fair",
4      "event": "Science Fair 2024",
5      "status": "Processing",
6      "location": "Exhibition Center - Hall A",
7      "date": "2024-09-05T09:00:00Z",
8      "checklist_id": "661631c15ca3f59d6",
9      "startTime": "2024-09-05T10:00:00Z",
10     "endTime": "2024-09-05T16:00:00Z",
11     "school_id": "661630a9577bed19d130",
12     "exhibition": "Local Innovators Show",
13     "note": "Extra chairs needed in the hall"
14 },
15 {
16     "id": "661633751a32e0d0bd170449",
17     "name": "Annual Science Fair",
18     "event": "Science Fair 2024",
19     "status": "Processing",
20     "location": "Exhibition Center - Hall A",
21     "date": "2024-09-05T09:00:00Z",
22     "checklist_id": "661e4fd9e977d7711c301a1",
23     "checklist": [
24         {
25             "id": "661e4fd9e977d7711c301a1",
26             "name": "Paid Workshop",
27             "task": [

```

```

27         },
28         "name": "Planner (from",
29         "link": "",
30         "status": 0
31     },
32     {
33         "name": "Make initial",
34         "link": "",
35         "status": 0
36     },
37     {
38         "name": "Note: Take F"]
39         "link": "",
40         "status": 0
41     },
42     {
43         "name": "If struggling",
44         "link": "",
45         "status": 0
46     },
47     {
48         "name": "Update holds",
49         "link": "https://apac-",
50         "status": 0
51     },
52     {
53         "name": "Add to LER &",
54         "link": "",
55         "status": 0
56     },
57     {
58         "name": "Respond with",
59         "link": "",
60         "status": 0
61     },
62     {
63         "name": "If Future Focu",
64         "link": "",
65         "status": 0
66     },
67     {
68         "name": "KE to roster",
69         "link": "",
70         "status": 0
71     },
72     {
73         "name": "Check in with",
74         "link": "",
75         "status": 0
76     }
77 ]
78 },
79 "startTime": "2024-09-05T10:00:00Z",
80 "endTime": "2024-09-05T16:00:00Z",
81 "school_id": "661630a9577bed19d130",
82 "school": {
83     "name": "Greenwood Elementary",
84     "studentYear": 2020,
85     "numStudentAttended": 150,
86     "numStudentRegistered": 200,

```

		<pre> 87 "hourRegistered": 1200, 88 "hourAttended": 1100, 89 "lowSES": true, 90 "allergy": "Peanuts", 91 "contactFirstName": "Jane", 92 "contactLastName": "Doe", 93 "email": "jane.doe@greenwood.org", 94 "phone": "555-1234", 95 "note": "", 96 "isAccessibility": false, 97 "isAllergy": false, 98 "isPartner": false 99 }, 100 "exhibition": "Local Innovators Showcase", 101 "note": "Extra chairs needed in the lot" 102 } </pre>	
POST	api/booking/	<pre> 1 { 2 "name": "Annual Science Fair", 3 "event": "Science Fair 2024", 4 "status": "Processing", 5 "location": "Exhibition Center - Hall B", 6 "date": "2024-09-05T09:00:00Z", 7 "checklist_id": "661e4fdae977d7711c301aff", 8 "startTime": "2024-09-05T10:00:00Z", 9 "endTime": "2024-09-05T16:00:00Z", 10 "school_id": "661630a9577bed19d130de77", 11 "exhibition": "Local Innovators Showcase", 12 "note": "Extra chairs needed in the lot", 13 "bus": { 14 "bus_req": false, 15 "isBooked": false, 16 "status": 0, 17 "price": 0.0, 18 "date_paid": null, 19 "invoice": null 20 }, 21 "per_student": 0, 22 "expense": 0, 23 "income": 0, 24 "profit": 0 25 } </pre>	checklist_id and school_id has to be valid
PUT	api/booking/6603d23b9e45d1f805c5fdd1/	<pre> 1 { 2 "name": "Annual Science Fair", 3 "event": "Science Fair 2024", 4 "status": "Success", 5 "location": "Exhibition Center - Hall B", 6 "date": "2024-09-05T09:00:00Z", 7 "checklist_id": "661e4fdae977d7711c301aff", 8 "startTime": "2024-09-05T10:00:00Z", 9 "endTime": "2024-09-05T16:00:00Z", 10 "school_id": "661630a9577bed19d130de77", 11 "exhibition": "Local Innovators Showcase", 12 "note": "Extra chairs needed in the lot", 13 "bus": { 14 "bus_req": false, 15 "isBooked": false, 16 "status": 0, 17 "price": 0.0, 18 "date_paid": null, 19 "invoice": null 20 }, 21 "per_student": 0, 22 "expense": 0, 23 "income": 0, 24 "profit": 0 25 } </pre>	

		<pre> 17 "price": 0.0 18 "date_paid": 19 "invoice": " 20 }, 21 "per_student": 0 22 "expense": 0, 23 "income": 0, 24 "profit": 0 25 }</pre>	
DELETE	api/booking/6603d23b9e45d1f805c5fdd1/		<pre> 1 { 2 "status": "success", 3 "id": "66049ef698b69f53ee92f823" 4 }</pre>

School

Method	URL Example	Request Body	Response	Description
GET	api/school/		<pre> 1 [2 { 3 "id": "661f69f37fedbcf1319be202", 4 "name": "Greenwood Elementary123213", 5 "studentYear": 2020, 6 "numStudentAttended": 150, 7 "numStudentRegistered": 200, 8 "hourRegistered": 1200, 9 "hourAttended": 1100, 10 "lowSES": true, 11 "allergy": "Peanuts", 12 "contactFirstName": "Jane", 13 "contactLastName": "Doe", 14 "email": "jane.doe@greenwood.edu", 15 "phone": "555-1234", 16 "note": "", 17 "isAccessibility": false, 18 "isAllergy": false, 19 "isPartner": false 20 } 21]</pre>	
GET	api/school/6603d23b9e45d1f805c5fdd1/		<pre> 1 { 2 "id": "661f69f37fedbcf1319be202", 3 "name": "Greenwood Elementary123213", 4 "studentYear": 2020, 5 "numStudentAttended": 150, 6 "numStudentRegistered": 200, 7 "hourRegistered": 1200, 8 "hourAttended": 1100, 9 "lowSES": true, 10 "allergy": "Peanuts", 11 "contactFirstName": "Jane", 12 "contactLastName": "Doe", 13 "email": "jane.doe@greenwood.edu", 14 "phone": "555-1234",</pre>	

			<pre> 15 "note": "", 16 "isAccessibility": false, 17 "isAllergy": false, 18 "isPartner": false 19 }</pre>	
POST	api/school/	<pre> 1 { 2 "name": "Greenwo 3 "studentYear": 2 4 "numStudentAttende 5 "numStudentRegiste 6 "hourRegistered": 7 "hourAttended": 8 "lowSES": true, 9 "allergy": "Peanu 10 "contactFirstName": 11 "contactLastName": 12 "email": "jane.d 13 "phone": "555-12 14 "note": "", 15 "isAccessibility": 16 "isAllergy": false 17 "isPartner": false 18 }</pre>	<pre> 1 { 2 "name": "Greenwood Elementary123", 3 "studentYear": 2020, 4 "numStudentAttended": 150, 5 "numStudentRegistered": 200, 6 "hourRegistered": 1200, 7 "hourAttended": 1100, 8 "lowSES": true, 9 "allergy": "Peanuts", 10 "contactFirstName": "Jane", 11 "contactLastName": "Doe", 12 "email": "jane.doe@greenwood.edu", 13 "phone": "555-1234", 14 "_id": "661e5111e977d7711c301b01" 15 }</pre>	
PUT	api/school/661 630a9577bed1 9d130de77/	<pre> 1 { 2 "name": "Greenwo 3 "studentYear": 2 4 "numStudentAttende 5 "numStudentRegiste 6 "hourRegistered": 7 "hourAttended": 8 "lowSES": true, 9 "allergy": "Peanu 10 "contactFirstName": 11 "contactLastName": 12 "email": "jane.d 13 "phone": "555-12 14 "note": "", 15 "isAccessibility": 16 "isAllergy": false 17 "isPartner": false 18 }</pre>	<pre> 1 { 2 "status": "success", 3 "id": "661e5111e977d7711c301b01" 4 "updated": 1 5 }</pre>	
DELETE	api/school/660 3d23b9e45d1f 805c5fdd1/		<pre> 1 { 2 "status": "success", 3 "id": "66049ef698b69f53ee92f823" 4 }</pre>	

Miscellaneous

Method	URL Example	Request Body	Response	Description
GET	api/miscellaneous/		<pre> 1 { 2 "id": "661f624bce65511a21305a41", 3 "module": [4 "Access Code", 5 "F: Bespoke", 6 "TBC", 7 "TOUR: Nadine Christensen - Facilitated", 8 "TOUR: Not Natural - Facilitated", 9 "VISIT: Ancient Lives - Self-directed", 10 "VISIT: Nadine Christensen - Self-Directed", 11 "W: App It!", 12 "W: Design It!", 13 "W: Future Food", 14 "W: Mission Control", 15 "W: Sustainable Communities", 16 "W: Take Flight" 17], 18 "program_stream": [19 "ART: Excursions", 20 "SCoE: Excursions", 21 "STEAM: Excursions", 22 "STEAM: Special Outreach Projects" 23], 24 "facilitators": [25 "EB", 26 "ER", 27 "JC", 28 "MC", 29 "MeLB", 30 "MK", 31 "Teacher Delivered", 32 "TS", 33 "XC" 34], 35 "delivery_location": [36 "Buxton", 37 "Embedded (25%)", 38 "Incursion", 39 "Old Quad", 40 "SGM: EG, WG", 41 "SGM: SGMT", 42 "SGM: W2", 43 "SGM: W3" 44], 45 "exhibition": [46 "Ancient Lives", 47 "Nadine Christensen", 48 "Non-Exhibition Linked", 49 "Not Natural" 50] 51 }</pre>	
POST	api/miscellaneous/	<pre> 1 { 2 "id": "661f624bce65511a21305a41", 3 "module": [4 "Access Code" </pre>	<pre> 1 { 2 "module": [3 "Access Code", 4 "F: Bespoke", </pre>	

		<pre> 5 "F: Bespoke" 6 "TBC", 7 "TOUR: Nadine Christensen - Facilitated" 8 "TOUR: Not Natural - Facilitated" 9 "VISIT: Ancient Lives - Self-directed" 10 "VISIT: Non-Exhibition Linked" 11 "W: App It!" 12 "W: Design It!" 13 "W: Future Food" 14 "W: Mission Control" 15 "W: Sustainable Communities" 16 "W: Take Flight" 17], 18 "program_stream": [19 "ART: Excursions", 20 "SCoE: Excursions", 21 "STEAM: Excursions", 22 "STEAM: Special Outreach Projects" 23], 24 "facilitators": [25 "EB", 26 "ER", 27 "JC", 28 "MC", 29 "MelB", 30 "MK", 31 "Teacher Delivered", 32 "TS", 33 "XC" 34], 35 "delivery_location": [36 "Buxton", 37 "Embedded (25%)", 38 "Incursion", 39 "Old Quad", 40 "SGM: EG, WG", 41 "SGM: SGMT", 42 "SGM: W2", 43 "SGM: W3" 44], 45 "exhibition": [46 "Ancient Lives", 47 "Nadine Christensen", 48 "Non-Exhibition Linked", 49 "Not Natural" 50] 51 } </pre>	<pre> 5 "TBC", 6 "TOUR: Nadine Christensen - Facilitated" 7 "TOUR: Not Natural - Facilitated" 8 "VISIT: Ancient Lives - Self-directed" 9 "VISIT: Nadine Christensen - Self-Directed" 10 "W: App It!", 11 "W: Design It!", 12 "W: Future Food", 13 "W: Mission Control", 14 "W: Sustainable Communities", 15 "W: Take Flight" 16], 17 "program_stream": [18 "ART: Excursions", 19 "SCoE: Excursions", 20 "STEAM: Excursions", 21 "STEAM: Special Outreach Projects" 22], 23 "facilitators": [24 "EB", 25 "ER", 26 "JC", 27 "MC", 28 "MelB", 29 "MK", 30 "Teacher Delivered", 31 "TS", 32 "XC" 33], 34 "delivery_location": [35 "Buxton", 36 "Embedded (25%)", 37 "Incursion", 38 "Old Quad", 39 "SGM: EG, WG", 40 "SGM: SGMT", 41 "SGM: W2", 42 "SGM: W3" 43], 44 "exhibition": [45 "Ancient Lives", 46 "Nadine Christensen", 47 "Non-Exhibition Linked", 48 "Not Natural" 49], 50 "_id": "661f624bce65511a21305a41" 51 } </pre>
--	--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

PUT	api/miscellaneous/	<pre> 1 { 2 "id": "661f624bce65511a21305a41" 3 "module": [4 "Access Codes" 5 "F: Bespoke" 6 "TBC", 7 "TOUR: Nadine Christensen - Facilitated" 8 "TOUR: Not Natural - Facilitated" 9 "VISIT: Ancient Lives - Self-directed" 10 "VISIT: Non-Exhibition Linked" 11 "W: App It!" 12 "W: Design It!" 13 "W: Future Food" 14 "W: Mission Control" 15 "W: Sustainable Communities" 16 "W: Take Flight" 17], 18 "program_stream": [19 "ART: Excursions", 20 "SCoE: Excursions", 21 "STEAM: Excursions", 22 "STEAM: Special Outreach Projects" 23], 24 "facilitators": [25 "EB", 26 "ER", 27 "JC", 28 "MC", 29 "MelB", 30 "MK", 31 "Teacher Delivered", 32 "TS", 33 "XC" 34], 35 "delivery_location": [36 "Buxton", 37 "Embedded (25%)", 38 "Incursion", 39 "Old Quad", 40 "SGM: EG, WG", 41 "SGM: SGMT", 42 "SGM: W2", 43 "SGM: W3" 44], 45 "exhibition": [46 "Ancient Lives", 47 "Nadine Christensen", 48 "Non-Exhibition Linked", 49 "Not Natural" 50], 51 "status": "success", 52 "updated": 1 53 } </pre>
-----	--------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

```
13         "W: Future F
14         "W: Mission
15         "W: Sustainab
16         "W: Take Fl
17     ],
18     "program_stream"
19     "ART: Excursu
20     "SCoE: Excurs
21     "STEAM: Excu
22     "STEAM: Specia
23 ],
24     "facilitators":
25     "EB",
26     "ER",
27     "JC",
28     "MC",
29     "MelB",
30     "MK",
31     "Teacher Devel
32     "TS",
33     "XC"
34 ],
35     "delivery_locati
36     "Buxton",
37     "Embedded (201
38     "Incursion",
39     "Old Quad",
40     "SGM: EG, W1"
41     "SGM: SGMT",
42     "SGM: W2",
43     "SGM: W3"
44 ],
45     "exhibition": [
46     "Ancient Lives"
47     "Nadine Christi
48     "Non-Exhibitio
49     "Not Naturalis
50   ]
51 }
```

TESTS

Test Directory Outline

I. Backend Testing

- **Location:** /src/back-end/api/tests
- **Purpose:** Validates the server-side APIs, ensuring they handle requests and return expected results for all CRUD operations.

1. Unit Test - GET

- **Description:** Tests all GET requests to fetch data from the server.
- **Test Cases Include:** - Fetching all records. - Fetching a single record by ID. - Fetching with queries and filters.

2. Unit Test - PUT

- **Description:** Tests PUT requests for updating existing data.
- **Test Cases Include:** - Updating records with valid data. - Handling updates with invalid or incomplete data.

3. Unit Test - DELETE

- **Description:** Tests DELETE requests to remove data.
- **Test Cases Include:** - Deleting a record by ID. - Ensuring records are not accessible post-deletion.

4. Unit Test - CREATE

- **Description:** Tests POST requests for creating new records.
- **Test Cases Include:** - Creating records with valid data. - Handling attempts to create records with invalid data.

II. Frontend Testing

- **Location:** /src/front-end/src/__tests__
- **Purpose:** Ensures that the user interface behaves as expected under various scenarios and interactions.

1. Unit Test - Dashboard

- **Description:** Tests the Dashboard component's responsiveness and functionality.
- **Test Cases Include:**
 - Rendering essential elements and UI components.
 - Interactions with filter options (e.g., type, date, location).
 - Sorting and displaying data based on user selections.

2. Unit Test - NewBooking

- **Description:** Tests the NewBooking component's ability to handle form inputs, validate data, and submit new bookings.
- **Test Cases Include:**
 - Switching between different booking types (Delivery, School, Bus, Others).
 - Correctly capturing and validating user inputs.
 - Submitting forms and handling the responses.

3. Unit Test - Templates

- **Description:** Verifies the functionality of the Templates component, focusing on its ability to list, add, and manage template items effectively.
- **Test Cases Include:**
 - Loading and displaying templates from a data fetch.

- Interactions with the UI to add new templates.
- Handling user actions such as selecting and viewing template details.

4. Unit Test - TemplateDetail

- **Description:** Focuses on testing the TemplateDetail component, ensuring it handles detailed interactions for creating and editing templates.
- **Test Cases Include:**
 - Loading and displaying individual template details.
 - Adding, updating, and deleting tasks within a template.
 - Validating user input and managing state transitions.
 - Reacting to user actions like save, delete, and discard changes.

Backend - Testing

Test Document: MongoDB Integration and API Response Test

1. Introduction

- This document outlines the test strategy for the `DBTest` class, which is responsible for validating the integration of MongoDB with a Django application and the correct responses of the application's REST API endpoints.

2. Scope

- This test plan covers the following:
 - Connection to the MongoDB database using the Django settings configuration.
 - Retrieval of data from specific API endpoints to ensure the expected responses are received for both successful and erroneous scenarios.

3. Objectives

- The main objectives of these tests are:
 - To confirm that the application can connect to and interact with MongoDB correctly.
 - To verify that the API responds with the correct status codes and data for various endpoints when accessed with correct and incorrect parameters.

4. Roles and Responsibilities

- **Test Engineer** (@Guixian Li) : Responsible for writing, executing, and maintaining the test cases.
- **Developer** (@Yuanbo Xu) : To review test failures and address any bugs or issues in the code.
- **Quality Assurance (one of the team members)**: Oversees the testing process and ensures quality standards.

5. Testing Strategy

5.1 Test Environment

- The tests will be executed in a controlled development environment with access to a MongoDB instance configured through Django settings.

5.2 Test Data

- Test data will include predefined identifiers for querying the API endpoints.
- Examples of valid data can be found in the team's GitHub repository  [LINKS & RESOURCES](#)

5.3 Types of Tests

- **Unit Tests**: To validate individual functions and API.
- Test Execution Schedule

The tests will be performed during the completion of every feature of the development cycle, before deployment. The expected duration of the testing phase is one week.

1. Risks and Contingencies

- **Risk**: Incorrect configuration of the database connection.
- **Mitigation**: Review and verify configuration settings before executing tests.
- **Risk**: Incomplete implementation of API endpoints.
- **Mitigation**: Coordinate with developers to ensure all endpoints are implemented before testing.

2. Defect Management

Defects found during testing will be logged into the project's issue tracker with severity, steps to reproduce, and screenshots if applicable. The development team will prioritize defects based on severity.

3. Metrics and Reporting

Test results will be compiled into a report showing:

- Number of tests executed.
- Number of tests passed/failed.
- Detailed logs for any failures.

4. Approval

Upon completion, the test report will be reviewed and approved by the rest of the team.

Coverage Score = 4/6

Frontend - Testing

Test Document for Dashboard Page (Frontend)

Overview

This document describes the frontend tests implemented to validate the functionality and UI of the Dashboard page within the application. These tests ensure the page correctly interacts with user inputs, properly filters and displays data, and responds with the expected UI changes.

Test Environment

- **Framework:** React
- **Testing Libraries:** `@testing-library/react`, `@testing-library/user-event`, `@testing-library/jest-dom`
- **Additional Tools:** React Router (BrowserRouter for navigation context)

Coverage Score = 4/6



CODE REVIEW

1. Frontend Code Review:

- How did the team perform **Frontend Section code review** - who participated in that, when did that happen, number of issues identified

2. Backend Code Review:

- How did the team perform **Backend Section code review** - who participated in that, when did that happen, number of issues identified

3. When to do Code Review:

- Whenever we want to do merge request.

4. ⚠ Important AI Feedback:

- Due to the technical problem, we can't receive the ai feedback from gpt model. It will be done in next sprint

5. Criteria for reviewing code:

- Functional Completeness
 - Ensure new features are completed as required.
 - Verify that the feature meets the user story or acceptance criteria.
- Code quality:
 - Follow coding standards and style guides.
 - Is the code easy to understand and maintain?
 - Use appropriate naming conventions.
 - Make sure there is no redundant or useless code
- Conflict Resolution
 - Before merging, resolve any code conflicts.
 - Ensure that the merged code executes without errors.
- Compatibility with existing functionality
 - Make sure new features don't break existing functionality.
- Performance
 - Check whether new features have a negative impact on application performance.
 - Assess performance bottlenecks and recommend optimisations.

Frontend Code Review

Total code review times: 13 (sprint 2)

Code Review Front-end #1 (Merge request #2)

Date: 28/03/2024

Merge Request From **Frontend/features/typography** to **Main**

Participants: [@Yun-Chi Hsiao](#) [@Yongli Qin](#) [@Daniel Su](#)

Check list:

- Check if the font family matches the design
- Check if the font size matches the design (Difference handing might use different font size)
- Check if the colour and contrast matches the design

Number of Issues : 0 (Standards compliant)

Action after code review: None

Code Review Front-end #2 (Merge request #2)

Date: 28/03/2024

Merge Request From **Frontend/features/navigationBar** to **Main**

Participants: [@Yun-Chi Hsiao](#) [@Yongli Qin](#) [@Daniel Su](#)

Details: Review the code of navigation bar written by [@Yongli Qin](#) and check if the layout and structure of the code is correct.

Check list:

- Make sure navigation items are clear and easy to understand.
- Verify that the navigation structure is logical and users can easily find the information they want.
- Validate style changes on mouseover, focus, and activation states to provide user feedback.
- The logo for each option is clear and intuitive

Number of Issues : 0 (Standards compliant)

Action after code review: None

Code Review Front-end #3 (Merge request #3)

Date: 09/04/2024

Merge Request From **Frontend/features/dashboard** to **Main**

Participants: [@Yun-Chi Hsiao](#) [@Yongli Qin](#) [@Daniel Su](#)

Check list:

- If the whole page is well structured.
- If the font size and font family is unified.
- Different buttons are available and the filter button works correctly.
- Hard code data is displayed correctly in the correct location

Number of Issues : 4

- There is a problem with the CSS of the dashboard page.
- The length of the divider is set too long.
- The page design needs to be improved.
- Code naming is not uniform

Action after code review:

- Redesign the dashboard page to make it more user friendly and modify the css.
- Re-naming the variables and match with the standard.
- Fix the problem of divider

Code Review Front-end #4 (Merge request #4)

Date: 09/04/2024

Merge Request From **Frontend/features/templates** to **Main**

Participants: [@Yun-Chi Hsiao](#) [@Yongli Qin](#) [@Daniel Su](#)

Check list:

- Check whether the function is completed
- Whether it conflicts with previous code.
- If the whole page is well structured.
- If the font size and font family is unified.
- Different buttons are available

Number of Issues : 0 (Standards compliant)

Action after code review: None

Code Review Front-end #5 (Merge request #5)

Date: 09/04/2024

Merge Request From **Frontend/features/login** to **Main**

Participants: [@Yun-Chi Hsiao](#) [@Yongli Qin](#) [@Daniel Su](#)

Check list:

- Check whether the function is completed
- Whether it conflicts with previous code.
- If the whole page is well structured.
- If the font size, font family and colour is unified.
- Different buttons are available

Number of Issues : 1

- inconsistent naming conventions

Action after code review:

- Fix the conventions of naming

Code Review Front-end #6 (Merge request #6)

Date: 10/04/2024

Merge Request From **Frontend/features/dashboard** to **Main**

Participants: [@Yun-Chi Hsiao](#) [@Yongli Qin](#) [@Daniel Su](#)

Check list:

- Check whether the function is completed
- Whether it conflicts with previous code.
- If the whole page is well structured.
- If the font size, font family and colour is unified.
- Different buttons are available
- Check if the previous bugs is fixed

Number of Issues : 0 (Standards compliant)

Action after code review: None

Code Review Front-end #7 (Merge request #7)

Date: 10/04/2024

Merge Request From **Frontend/features/login** to **Main**

Participants: [@Yun-Chi Hsiao](#) [@Yongli Qin](#) [@Daniel Su](#)

Check list:

- Check whether the function is completed
- Whether it conflicts with previous code.
- If the whole page is well structured.
- If the font size, font family and colour is unified.
- Different buttons are available
- Check if the previous bugs is fixed

Number of Issues : 0 (Standards compliant)

Action after code review: None

Code Review Front-end #8 (Merge request #9)

Date: 10/04/2024

Merge Request From **Frontend/features/fix** to **Main**

Participants: [@Yun-Chi Hsiao](#) [@Yongli Qin](#) [@Daniel Su](#)

Check list:

- Check if the previous bugs is fixed

Number of Issues : 1

- Navigation Bar still contains minor bugs, when some of the icon is clicked, the colour should be changed

Action after code review:

- Address the existing bugs and verify whether there are any additional potential issues in the code.

Code Review Front-end #9 (Merge request #10)

Date: 10/04/2024

Merge Request From **Frontend/features/fix** to **Main**

Participants: [@Yun-Chi Hsiao](#) [@Yongli Qin](#) [@Daniel Su](#)

Check list:

- Check if the previous bugs is fixed

Number of Issues : 0 (Standards compliant)

Action after code review: None

Code Review Front-end #10 (Merge request #12)

Date: 13/04/2024

Merge Request From **Frontend/features/templates-details** to **Main**

Participants: [@Yun-Chi Hsiao](#) [@Yongli Qin](#) [@Daniel Su](#)

Check list:

- Check whether the function is completed
- Whether it conflicts with previous code.
- If the whole page is well structured.
- If the font size, font family and colour is unified.
- Different buttons are available

Number of Issues : 0 (Standards compliant)

Action after code review: None

Code Review Front-end #11 (Merge request #13)

Date: 13/04/2024

Merge Request From **Frontend/features/newbooking** to **Main**

Participants: [@Yun-Chi Hsiao](#) [@Yongli Qin](#) [@Daniel Su](#)

Check list:

- Check whether the function is completed
- Whether it conflicts with previous code.
- If the whole page is well structured.
- If the font size, font family and colour is unified.
- Different buttons are available

Number of Issues : 1

- Inconsistent naming conventions

Action after code review:

- Fix the conventions of naming

Code Review Front-end #12 (Merge request #15)

Date: 13/04/2024

Merge Request From **Frontend/features/fix** to **Main**

Participants: [@Yun-Chi Hsiao](#) [@Yongli Qin](#) [@Daniel Su](#)

Check list:

- Check if the previous bugs is fixed

Number of Issues : 0 (Standards compliant)

Action after code review: None

Code Review Front-end #13 (Merge request #17)

Date: 13/04/2024

Merge Request From **Frontend/features/template-api** to **Main**

Participants: [@Yun-Chi Hsiao](#) [@Yongli Qin](#) [@Daniel Su](#)

Check list:

- Check whether the function is completed
- Whether it conflicts with previous code.
- If the whole page is well structured.
- If the font size, font family and colour is unified.
- Different buttons are available

Number of Issues : 0 (Standards compliant)

Action after code review: None

Code Review Front-end #14 (Merge request #29)

Date: 01/05/2024

Merge Request From **Frontend/features/dashboardAPI** to **Main**

Participants: [@Yun-Chi Hsiao](#) [@Yongli Qin](#) [@Daniel Su](#)

Check list:

- Check whether the function is completed
- Check if the api can work properly
- Check if the booking will appear on dashboard after creating
- Check if all the information is recorded and uploaded

Number of Issues : 0 (Standards compliant)

Action after code review: None

Backend Code Review

Total code review times: 7 (sprint 2)

Code Review Back-end #1 (Merge request #8)

Date: 10/04/2024

Merge Request From **backend/features/rest-api** to **Main**

Participants: [@Guixian Li](#) [@Yuanbo Xu](#) [@Pangfeng ZHENG](#)

Check list:

- Review the code of rest-api written by [@Yuanbo Xu](#)
- Use postman to check if the api is working properly and return the expected result.

Number of Issues : 0 (Standards compliant)

Action after code review: None

Code Review Back-end #2 (Merge request #11)

Date: 10/04/2024

Merge Request From **backend/features/rest-api** to **Main**

Participants: [@Guixian Li](#) [@Yuanbo Xu](#) [@Pangfeng ZHENG](#)

Check list:

- Review the code of db-connection written by [@Pangfeng ZHENG](#)
- Use postman to check if the api is working properly and return the expected result.

Number of Issues : 0 (Standards compliant)

Action after code review: None

Code Review Back-end #3 (Merge request #14)

Date: 17/04/2024

Merge Request From **backend/features/db-unit-test** to **Main**

Participants: [@Guixian Li](#) [@Yuanbo Xu](#) [@Pangfeng ZHENG](#)

Check list:

- Review the code of db-unit-test written by [@Guixian Li](#)
- Check if the unit test is working and produce useful information.

Number of Issues : 0 (Standards compliant)

Action after code review: None

Code Review Back-end #4 (Merge request #16)

Date: 17/04/2024

Merge Request From **backend/features/views** to **Main**

Participants: @Guixian Li @Yuanbo Xu @Pangfeng ZHENG

Check list:

- Review the code of views written by @Yuanbo Xu
- check if all the serializer is generated correct

Number of Issues : 0 (Standards compliant)

Action after code review: None

Code Review Back-end #5 (Merge request #21)

Date: 23/04/2024

Merge Request From **backend/features/views** to **Main**

Participants: @Guixian Li @Yuanbo Xu @Pangfeng ZHENG

Check list:

- Check if the chatgpt code review can work properly.

Number of Issues : 0 (Standards compliant)

Action after code review: None

Code Review Back-end #6 (Merge request #27)

Date: 24/04/2024

Merge Request From **backend/features/deployment** to **Main**

Participants: @Guixian Li @Yuanbo Xu @Pangfeng ZHENG

Check list:

- Check if deployment can work
- Check if the link is active

Number of Issues : 0 (Standards compliant)

Action after code review: None

Code Review Back-end #7 (Merge request #28)

Date: 24/04/2024

Merge Request From **backend/features/database** to **Main**

Participants: @Guixian Li @Yuanbo Xu @Pangfeng ZHENG @Daniel Su

Check list:

- Update the database schema
- Make sure the previous database model can work properly

Number of Issues : 0 (Standards compliant)

Action after code review: None

Front-end check list before doing the merge request

Functional Completeness

- Confirm that new features are fully implemented as required.
- Verify that the feature aligns with the user story or meets the acceptance criteria.

Code Quality

- Ensure adherence to front-end coding standards and style guides.
- Assess if the code is easy to understand and maintain.
- Check for appropriate use of naming conventions.
- Ensure CSS styles are consistent and adhere to design specifications.
- Redundant or unnecessary code has been removed.

Compatibility

- New features perform consistently across different browsers and devices.
- Ensure new features do not disrupt existing front-end functionality.

Performance

- Analyze if new features affect the loading and execution speed of the front-end application.
- Identify and optimize front-end performance bottlenecks, such as image optimization, asynchronous loading, etc.

Back-end check list before doing the merge request

Functional Completeness

- Confirm that new features are fully implemented as required.
- Verify that the functionality aligns with business logic and interface documentation.

Code Quality

- Adhere to back-end coding standards and style guides.
- Assess if the code is easy to understand and maintain.
- Check for appropriate use of naming conventions.
- Redundant or unnecessary code has been removed.

Data Handling

- Ensure data processing logic is correct and interactions with the database are efficient.
- Ensure data security and privacy measures are in place.

Performance

- Analyze if new features affect the processing and response times of the back-end application.
- Identify and optimize back-end performance bottlenecks, such as database query optimizations, caching mechanisms, etc.



LINKS & RESOURCES

- **Trello** [COMP90082_2024_SM1_SG_KOALA](#)
- **GitHub** <https://github.com/COMP90082-2024-SM1/SG-Koala> Connect your Github account
- **Figma** [Booking Management System](#)
- **Priava API**
 - <https://priavasupportcenter.ungerboeck.com/hc/en-us/categories/10843507387287-API-Integrations>

 Demo

Link to demonstration video:

<https://www.youtube.com/watch?v=H5LxiF4b4GY&feature=youtu.be>

