

SOFTWARE DESIGN DOCUMENT (SDD)




META EVENT APPLICATION

DES3043: REKA BENTUK PERISIAN (A)

NAME	MATRIC NUMBER
WONG CHUN LIANG	D20231106337
GOH QUO TENG	D20231106372

DR. ROZNIM BINTI MOHAMAD RASLI

	DOCUMENT IDENTIFICATION	
	APPLICATION NAME	VERSION
	META EVENT APPLICATION	3.0

REVISION HISTORY & DOCUMENT APPROVAL

REVISION HISTORY

DATE	VERSION	AUTHOR	REVISION DETAILS
15.6.2025	Version 1.0	WONG CHUN LIANG	First version created
21.6.2025	Version 2.0	WONG CHUN LIANG	Second version created
30.6.2025	Version 3.0	WONG CHUN LIANG	Third version created

DOCUMENT APPROVAL

The following Software Requirements and Specification has been accepted and approved by the following:

SIGNATURE	NAME	DATE
<i>WONG</i>	WONG CHUN LIANG	30.6.2025
	DR. ROZNIM BINTI MOHAMAD RASLI	


	DOCUMENT IDENTIFICATION	
	APPLICATION NAME	VERSION
	META EVENT APPLICATION	3.0

TABLE OF CONTENT

1.0 INTRODUCTION.....	3
1.1 Purpose.....	3
1.2 Scope.....	3
1.3 Definitions, Acronyms, and Abbreviations.....	3
1.4 References.....	4
1.5 Overview.....	4
2.0 PROPOSED SOFTWARE ARCHITECTURE.....	5
2.1 View 1 (ADV 1).....	5
2.2 View 2 (ADV 2).....	7
3.0 SUBSYSTEM DECOMPOSITION AND DESIGN GOALS.....	8
3.1 Subsystem Decomposition.....	8
3.1.1 Subsystem Venue Booking (WONG CHUN LIANG).....	8
3.1.2 Subsystem CHAMP Point Management (GOH QUO TENG).....	14
3.2 Design Goals.....	20
3.3 Addressing Design Goal.....	22
4.0 DETAILED DESIGN.....	28
4.1 Navigation Design.....	28
4.1.1 META Event Application Navigation Design.....	28
4.1.1.1 Subsystem Venue Booking (WONG CHUN LIANG).....	28
4.1.1.2 Subsystem CHAMP Point Management (GOH QUO TENG).....	35
4.2 Content Table.....	41

1.0 INTRODUCTION

This section provides a purpose of this document and includes a compilation of abbreviations and definitions to enhance clarity and comprehension. It also offers an overview and scope description of the contents outlined in this Software Design Document (SDD).

1.1 Purpose

The purpose of this Software Design Document (SDD) is to provide detailed information of the designs of all functions and specifications for the Meta Event Application. It explains the software and hardware components, detailed processes and data structures. This document also consists of architectural design, design diagram for user interface and detailed design with description

1.2 Scope

META Event : Organiser of META Faculty is a application designed to empower the META Faculty in organizing events effortlessly, featuring capabilities such as paperwork submission, event committee management , venue booking, CHAMP points management, event registration.

1.3 Definitions, Acronyms, and Abbreviations

Table 1.1: Definitions, Acronyms, and Abbreviations of META Event :
Organiser of META Faculty

Term	Definition
META Event	Name of the application. The word 'META' comes from META Faculty and 'Event' refers to the event that is being organized.
Organiser	This word brings the definition of the role as an helpful organiser to manage an event.
META Faculty	The application is designed to assist META Faculty.

1.4 References

1. Ali, H., Tan, C. K., Ibrahim, M. S., Lee, W. Y., & Siti, N. R. (2023). *Requirement Gathering Data for Meta Event Application: Comprehensive User Insights*. Unpublished Raw Data.
2. Ali, H., Tan, C. K., Ibrahim, M. S., Lee, W. Y., & Siti, N. R. (2024). *Requirement Modeling Data & Framework for Meta Event Application: Enhanced User Experience*. Unpublished Raw Data.

1.5 Overview

The remaining section of this document continues with 2 chapters which covers the different parts of the design for the Meta Event Application. Chapter 2 in this document provides three different software architectural diagrams which are ADV 1 (View 1), and ADV 2 (View 2) with their detailed descriptions. This will help the stakeholders understand the application and its main uses and related architecture in more detail. Chapter 3 provides all the application's design goals, subsystems and decomposition table with explicit definitions. This will clarify the design goals of Meta Event Application information and their targeted goals for the application. Chapter 4 will explicitly show the various interfaces related to Meta Event Application as well as many different analytical diagrams and models. This will be defined and explained to help stakeholders understand the underlying.

2.0 PROPOSED SOFTWARE ARCHITECTURE

2.1 View 1 (ADV 1)

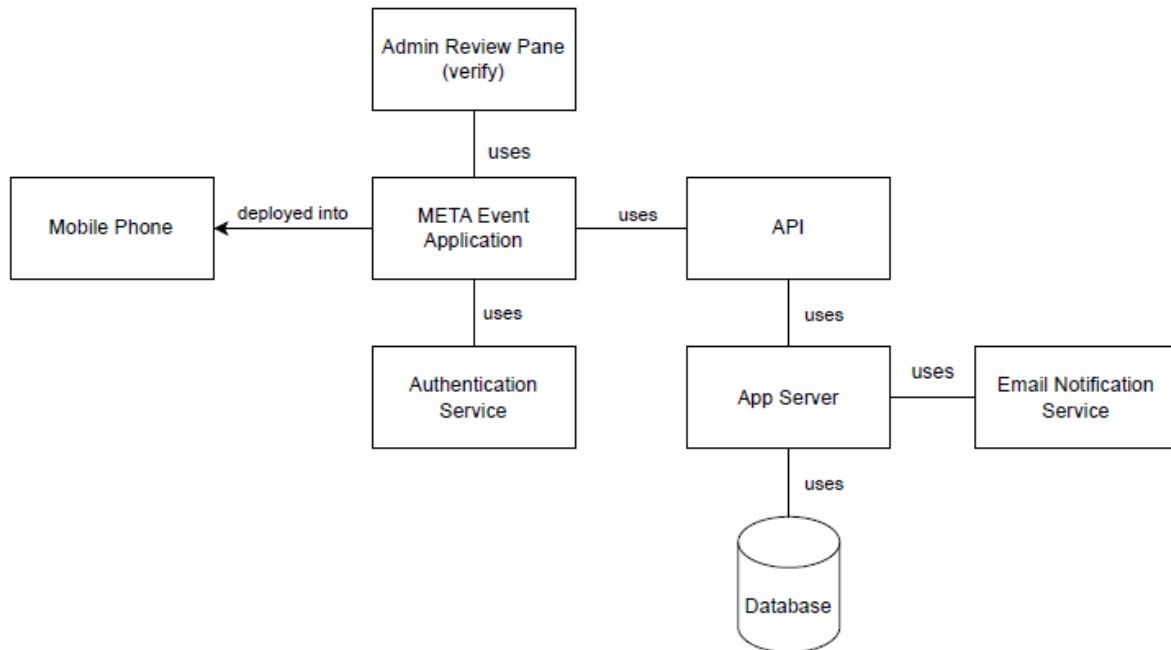


Diagram 1 : Architecture Design View (ADV 1)

The Architecture Design View (ADV 1) of the META Event Application illustrates how the system’s core components interact across various user roles and platforms to deliver a seamless event management experience for UPSI students and administrators.

At the front end, the application is deployed onto mobile devices, enabling students and admins to access the platform on the go. Through this mobile interface, users can register for venue, and track CHAMP points. This ensures convenient, role-based access to the application’s key features.

At the heart of the system is the META Event Application, which acts as the central hub coordinating all modules. The system incorporates a Function Venue Booking module that handles the reservation of event spaces, and is connected to an Admin Review Pane, which is used to verify submissions and bookings. These features ensure efficient administrative oversight.

The system is also integrated with the CHAMP Point Management module, which tracks students’ participation and awards CHAMP points accordingly. This module connects to a

shared API layer, which handles all communication between the application and the backend services.

The API communicates with the Application Server, which manages the application logic and interfaces with the Database. The database stores critical information including venue availability, CHAMP points, and event details. This layered backend architecture ensures scalability and data consistency. Additionally, the application integrates with an external Email Service to notify users and administrators about booking confirmations or approvals. The App Server communicates with this external service to trigger automated email notifications in real time.

All modules and components are interlinked and rely on secure and authenticated access as enforced through role-based login mechanisms. Real-time updates such as venue availability and CHAMP point totals are supported via WebSocket or equivalent technologies, while administrative actions trigger email notifications for verification outcomes.

In summary, the ADV 1 diagram showcases how different modules of the META Event Application work together in a distributed architecture, ensuring a responsive, modular, and role-oriented system for managing university events effectively.

2.2 View 2 (ADV 2)

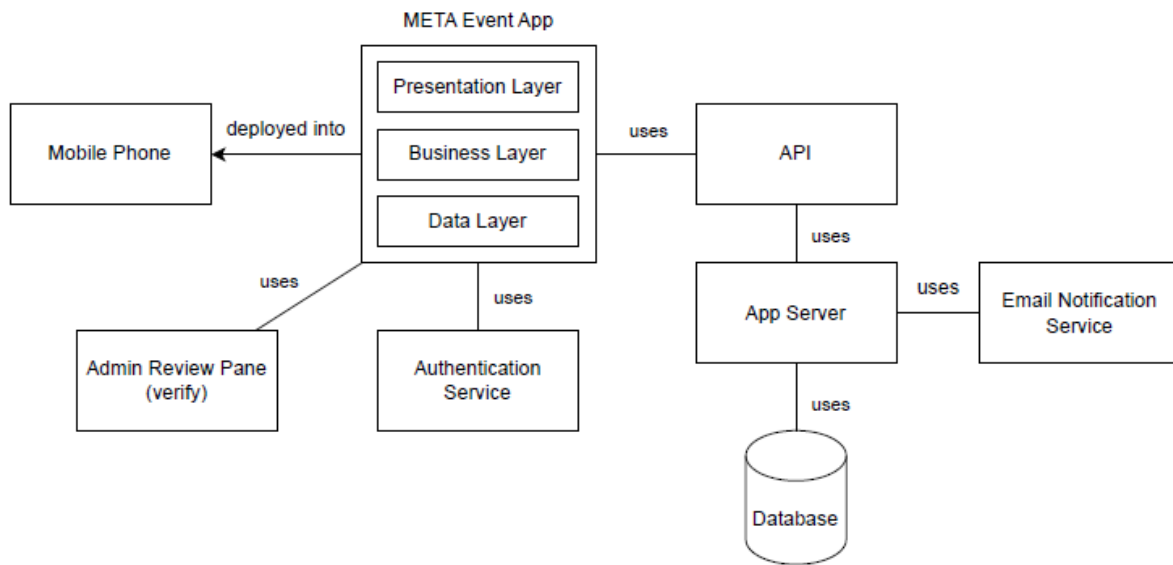


Diagram 2 : Architecture Design View (ADV 2)

The Architecture Design View (ADV 2) of the META Event Application outlines a three-layered architecture—Presentation, Business, and Data layers—deployed across web and mobile platforms to support event management functions. The system is accessible via mobile phones and connects to core modules such as Venue Booking, Admin Review Pane, and CHAMP Point Management. Admins can verify venue applications through the Admin Review Pane, which interfaces with the Function Venue Booking module. All interactions are processed through API that connect to an application server and centralized database, ensuring real-time updates and data consistency. The architecture supports secure role-based access, enabling students and admins to perform tasks such as venue booking and champ point tracking. This layered and modular structure promotes scalability, efficient communication, and smooth user experiences for both students and administrators within the META Faculty.

3.0 SUBSYSTEM DECOMPOSITION AND DESIGN GOALS

3.1 Subsystem Decomposition

3.1.1 Subsystem Venue Booking (WONG CHUN LIANG)

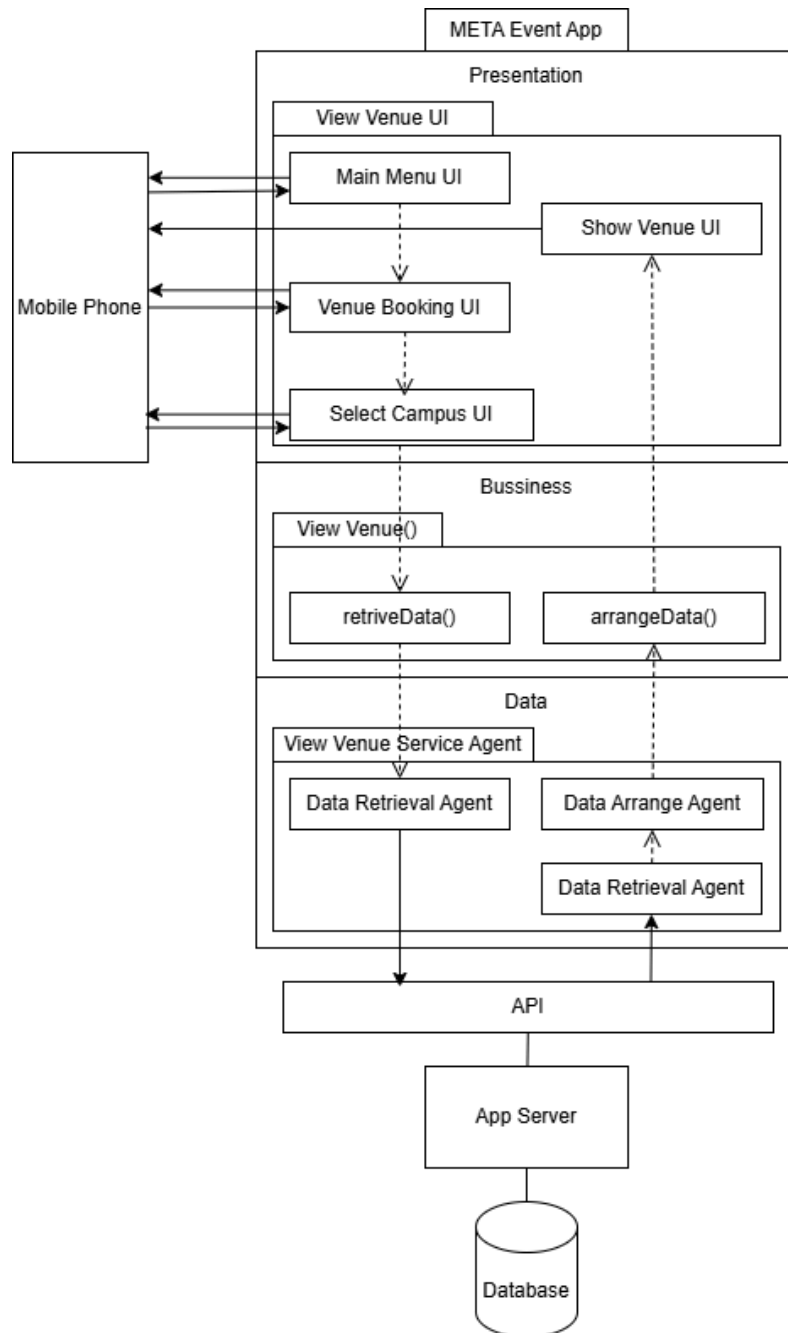


Diagram 3 : Subfunction View Venue in Venue Booking

This View Venue Subfunction allows both META students to view the name of the venue, the status of a venue (whether it's available for booking or not) and the size (number of seats) of the venue.

Subsystem	Function	Presentation	Business	Data
Venue Booking Subsystem	View Venue	<p>View Venue UI Package</p> <p>Component P1.1 Main Menu UI</p> <p>Component P1.2 Venue Booking UI</p> <p>Component P1.3 Select Campus UI</p> <p>Component P1.4 Show Venue UI</p>	<p>View Venue() Package</p> <p>Component B1.1 retrieveData()</p> <p>Component B1.2 arrangeData()</p>	<p>View Venue Service Agent Package</p> <p>Component D1.1 Data Retrieval Agent</p> <p>Component D1.2 Data Arrange Agent</p>

Table 1 : Subfunction View Venue in Venue Booking

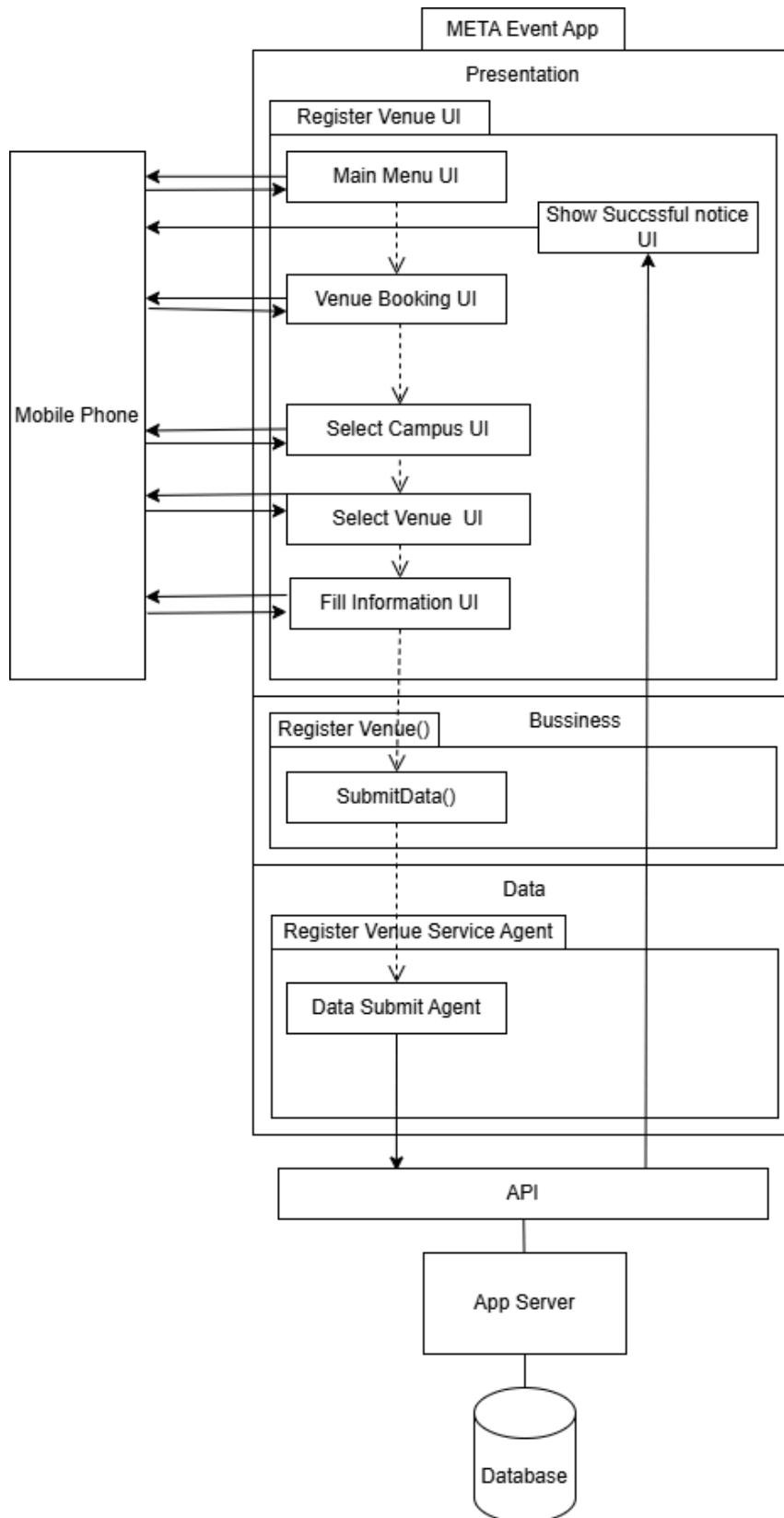


Diagram 4 : Subfunction Register Venue in Venue Booking

This Register Venue Subfunction allows META students to request a venue booking by specifying their preferred location and times.

Subsystem	Function	Presentation	Business	Data
Venue Booking Subsystem	Register Venue	<p>Register Venue UI Package</p> <p>Component P2.1 Main Menu UI</p> <p>Component P2.2 Venue Booking UI</p> <p>Component P2.3 Select Campus UI</p> <p>Component P2.4 Select Venue UI</p> <p>Component P2.5 Fill Information UI</p> <p>Component P2.6 Show Successful Notice UI</p>	<p>Register Venue() Package</p> <p>Component B2.1 submitData()</p>	<p>Register Venue Service Agent Package</p> <p>Component D2.1 Data Submit Agent</p>

Table 2 : Subfunction Register Venue in Function Venue Booking

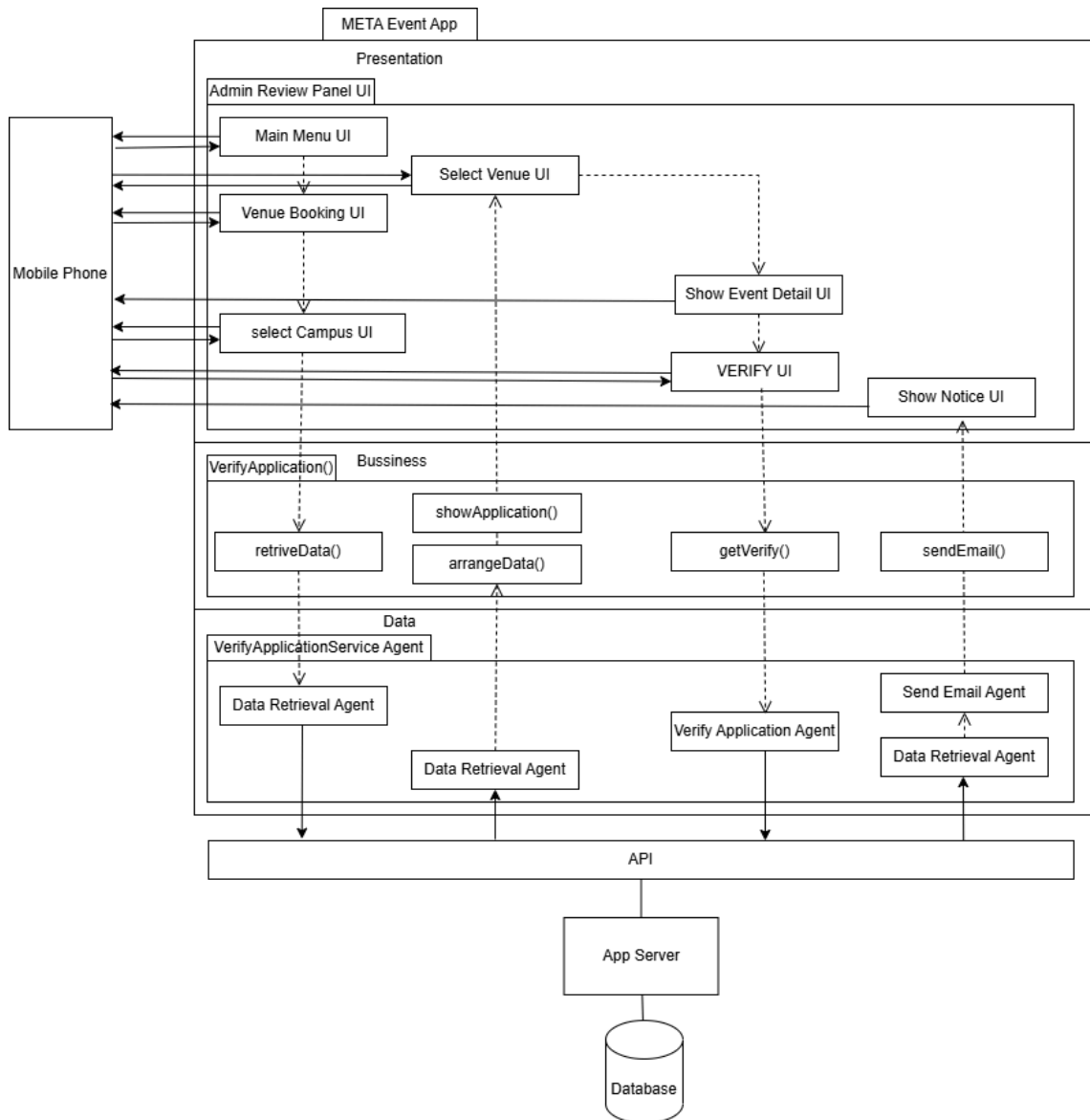


Diagram 5 : Subfunction Verify Venue Booking Application in Function Venue Booking

This Verify Venue Booking Application Subfunction allows Admin to review and verify these booking requests.

Subsystem	Function	Presentation	Business	Data
Venue Booking Subsystem	Verify Venue Booking Application	Admin Review Panel UI Package Component P3.1 Main Menu UI Component P3.2 Venue Booking UI Component P3.3 Select Campus UI Component P3.4 Select Venue UI Component P3.6 Show Event Detail UI Component P3.7 Verify UI Component P3.8 Show Notice UI	VerifyAppication() Package Component B3.1 retriveData() Component B3.2 arrangeData() Component B3.3 showApplication () Component B3.4 getVerify () Component B3.5 sendEmail ()	Verify Application Service Agent Package Component D3.1 Data Retrieval Agent Component D3.2 Data Retrieval Agent Component D3.3 Verify Application Agent Component D3.4 Data Retrieval Agent Component D3.5 Send Email Agent

Table 3 : Subfunction Verify Venue Booking Application in Function Venue Booking

3.1.2 Subsystem CHAMP Point Management (GOH QUO TENG)

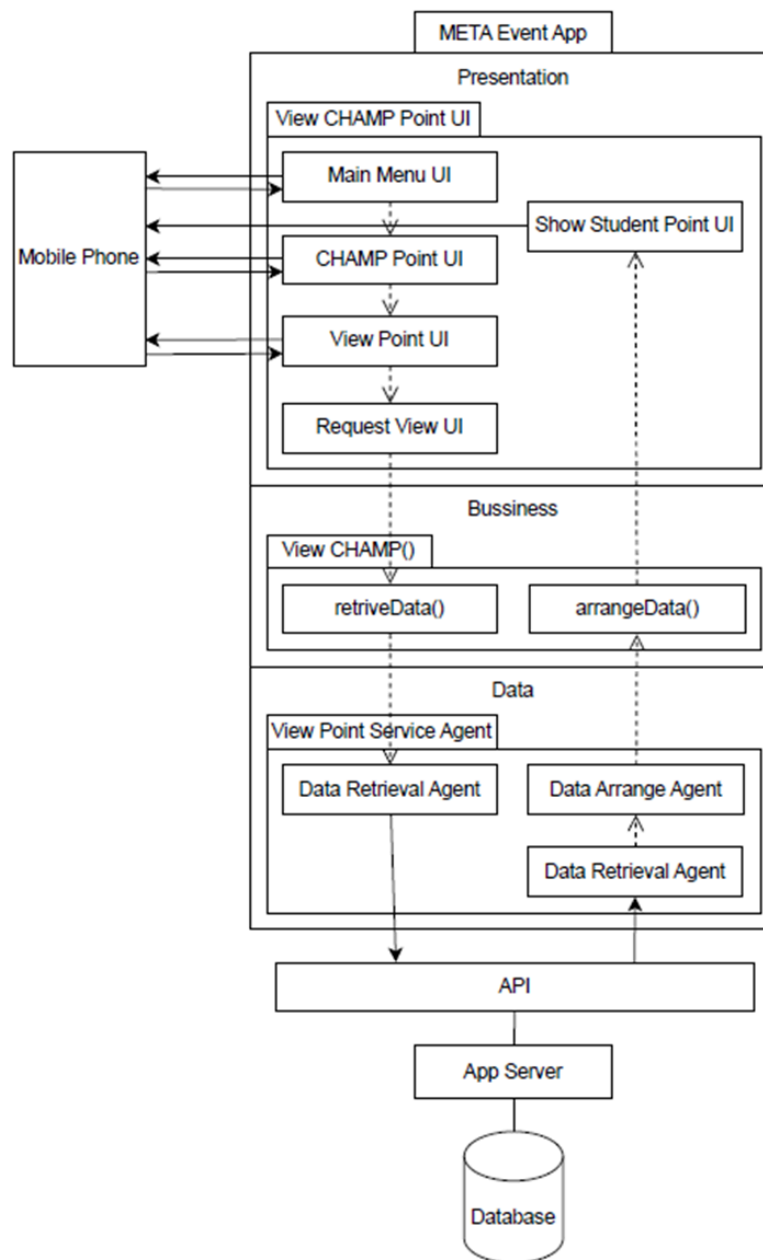


Diagram 6 : Subfunction View CHAMP Point in CHAMP Point Management

This is the View CHAMP Point subfunction in the CHAMP Point Management Subsystem. This subfunction allows META students to view and check the CHAMP points and total CHAMP points they have obtained by participating in events. And this subfunction also allows admin to view all META student’s CHAMP points.

Subsystem	Function	Presentation	Business	Data
CHAMP Point Management Subsystem	View CHAMP Point	View CHAMP Point UI Package Component P4.1 Main Menu UI Component P4.2 CHAMP Point UI Component P4.3 View Point UI Component P4.4 Request View UI Component P4.5 Show Student Point UI	View CHAMP() Package Component B4.1 retrieveData() Component B4.2 arrangeData()	View Point Service Agent Package Component D4.1 Data Retrieval Agent Component D4.2 Data Arrange Agent

Table 4 : Subfunction View CHAMP Point in CHAMP Point Management

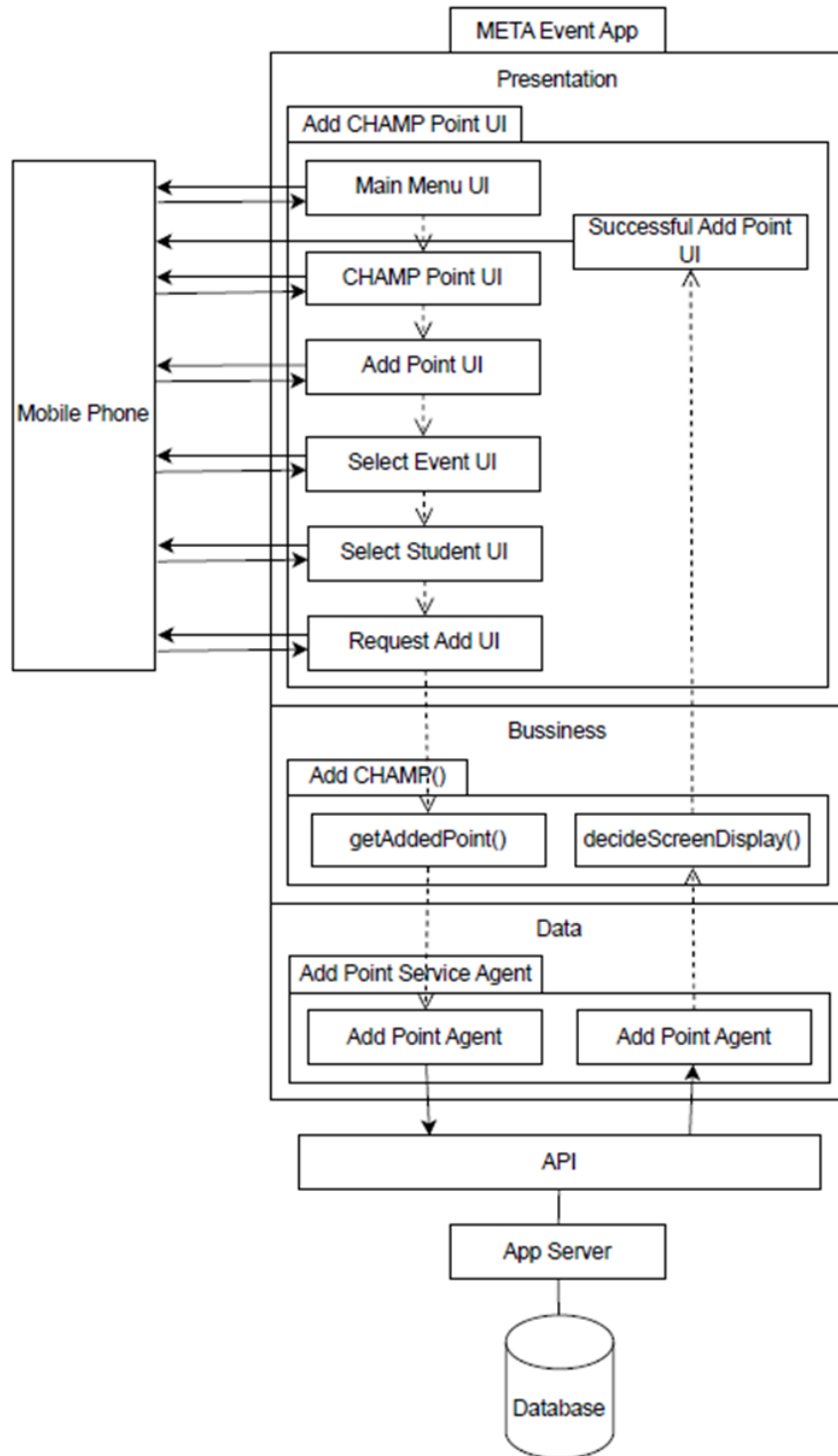


Diagram 7 : Subfunction Add CHAMP Point in CHAMP Point Management

This is the Add CHAMP Point subfunction in CHAMP Point Management Subsystem. This subfunction can allow admin to add the CHAMP points for all META students participating in the event.

Subsystem	Function	Presentation	Business	Data
CHAMP Point Management Subsystem	Add CHAMP Point	<p>Add CHAMP Point UI Package</p> <p>Component P5.1 Main Menu UI</p> <p>Component P5.2 CHAMP Point UI</p> <p>Component P5.3 Add Point UI</p> <p>Component P5.4 Select Event UI</p> <p>Component P5.5 Select Student UI</p> <p>Component P5.6 Request Add UI</p> <p>Component P5.7 Successful Add Point UI</p>	<p>Add CHAMP() Package</p> <p>Component B5.1 getAddedPoint()</p> <p>Component B5.2 decideScreenDisplay()</p>	<p>Add Point Service Agent Package</p> <p>Component D5.1 Add Point Agent</p>

Table 5 : Subfunction Add CHAMP Point in CHAMP Point Management

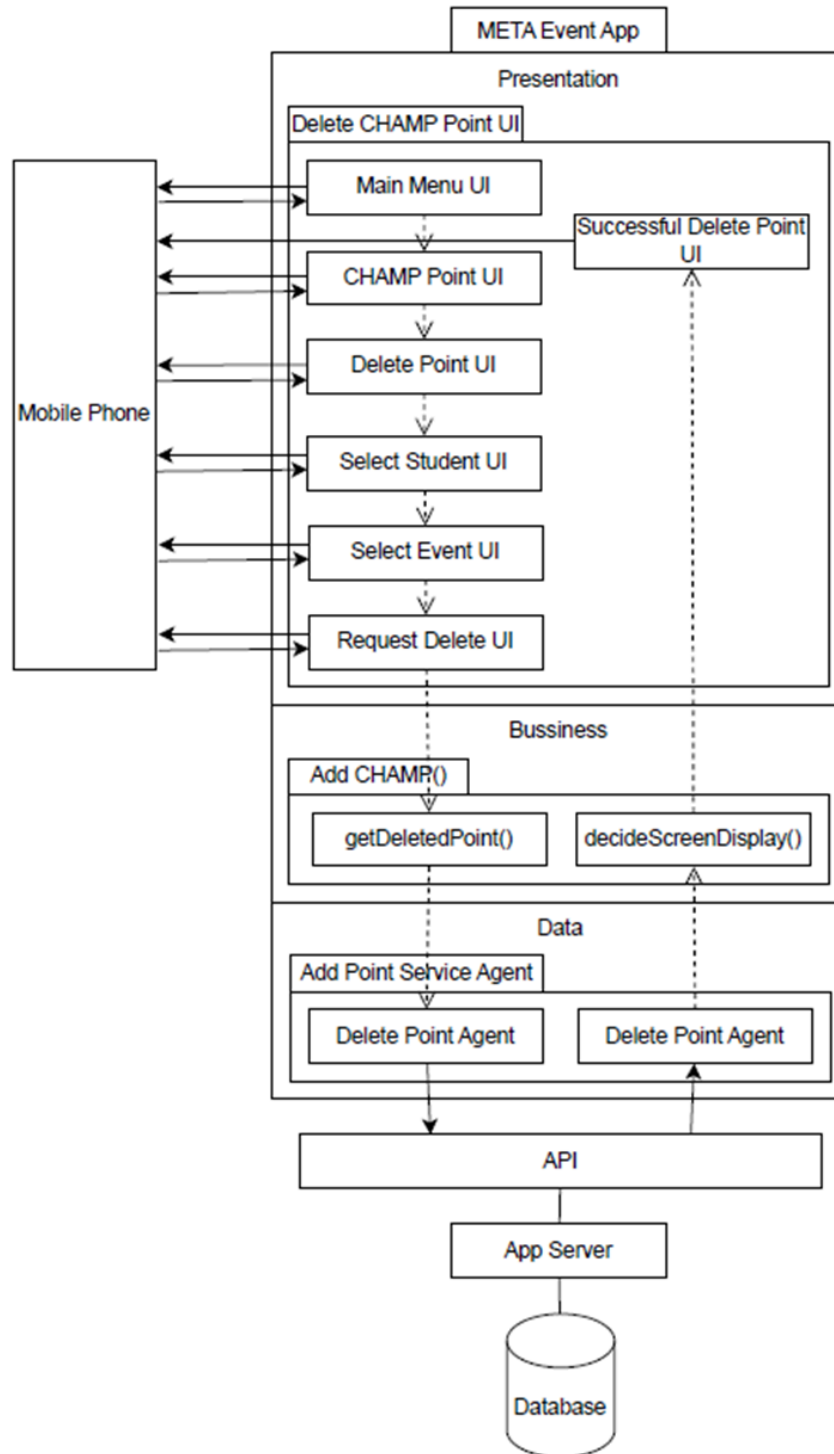


Diagram 8 : Subfunction Delete CHAMP Point in CHAMP Point Management

This is the Delete CHAMP Point subfunction in CHAMP Point Management subsystem. This subfunction can allow admin to delete the CHAMP points that were received by the META student through participating events.

Subsystem	Function	Presentation	Business	Data
CHAMP Point Management Subsystem	Delete CHAMP Point	<p>Delete CHAMP Point UI Package</p> <p>Component P6.1 Main Menu UI</p> <p>Component P6.2 CHAMP Point UI</p> <p>Component P6.3 Delete Point UI</p> <p>Component P6.4 Select Student UI</p> <p>Component P6.5 Select Event UI</p> <p>Component P6.6 Request Delete UI</p> <p>Component P6.7 Successful Delete Point UI</p>	<p>Delete CHAMP() Package</p> <p>Component B6.1 getDeletedPoint()</p> <p>Component B6.2 decideScreenDisplay()</p>	<p>Delete Point Service Agent Package</p> <p>Component D6.1 Delete Point Agent</p>

Table 6 : Subfunction Delete CHAMP Point in CHAMP Point Management

3.2 Design Goals

Design Goals	QR_ID	Quality Requirements	Design Criterion Group	Design Criterion	Subsystem Involved
Design Goal 1	QR001	The application should provide an intuitive user interface, ensuring ease of navigation for META students and admin.	Usability	Ease To Use	Subsystem Venue Booking Subsystem CHAMP Point Management
	QR002	All data exchanges should be encrypted, and role-based access controls should prevent unauthorized actions.	Security	Access Control	Subsystem Venue Booking Subsystem CHAMP Point Management
Design Goal 2	QR003	Actions such as login, submission, and data retrieval should complete within 2 seconds under standard conditions.	Performance	Response Time	Subsystem Venue Booking Subsystem CHAMP Point Management

	QR004	Interactive buttons should be responsive and execute actions immediately when tapped	Performance	Response Time	Subsystem Venue Booking Subsystem CHAMP Point Management
Design Goal 3	QR005	The application should support an increasing number of users and events without requiring major architectural changes.	Scalability	System Scalability	Subsystem Venue Booking Subsystem CHAMP Point Management
	QR006	The system should display real-time updates on venue and people availability.	Performance	Real-Time Feedback	Subsystem Venue Booking
	QR007	Data related to events, committee members, and champ points should be accurate and consistent across all modules.	Data Quality	Data Consistency	Subsystem CHAMP Point Management

Table 7 : Table of Design Goals

3.3 Addressing Design Goal

Design Goals	QR_ID	Quality Requirements	Design Criterion Group	Design Criterion	Subsystem Involved
Design Goal 1	QR001	The application should provide an intuitive user interface, ensuring ease of navigation for META students and admin.	Usability	Ease To Use	Subsystem Venue Booking Subsystem CHAMP Point Management
<p>Issues:</p> <ol style="list-style-type: none"> 1. Mapping Subsystem to Processors and Components UI components must be lightweight and efficiently distributed to client-side (mobile/web). 2. Identifying and Storing Persistent Data Minimal; only user settings or interface preferences if needed. 3. Providing Access Control Ensure UI adapts based on user role (e.g., admin sees more options than students). 4. Designing the Global Control Flow Should allow smooth navigation between modules without re-authentication or page reloads. 5. Identifying Boundary Condition Handle screen resolution, loading errors, or missing data gracefully. 					

	QR002	All data exchanges should be encrypted, and role-based access controls should prevent unauthorized actions.	Security	Access Control	Subsystem Venue Booking Subsystem CHAMP Point Management
<p>Issues:</p> <ol style="list-style-type: none"> 1. Mapping Subsystem to Processors and Components Use secure APIs, encryption handled at both client and server endpoints. 2. Identifying and Storing Persistent Data Encrypt sensitive user and event data at rest and in transit. 3. Providing Access Control Role-based authentication and authorization; use JWT, OAuth, or RBAC systems. 4. Designing the Global Control Flow Redirect unauthorized users or restrict access to certain processes dynamically. 5. Identifying Boundary Condition Fail securely (e.g., reject access with proper error messages, log attempts). 					
Design Goal 2	QR003	Actions such as login, submission, and data retrieval should complete within 2 seconds under standard conditions.	Performance	Response Time	Subsystem Venue Booking Subsystem CHAMP Point Management

	<p>Issues:</p> <ol style="list-style-type: none"> 1. Mapping Subsystem to Processors and Components Optimize database access and server processing; consider load balancing. 2. Identifying and Storing Persistent Data Use caching for frequently accessed data (e.g., event list, user profiles). 3. Providing Access Control Minimal effect; ensure access checks don't slow down the response. 4. Designing the Global Control Flow Asynchronous loading of data to reduce perceived wait times. 5. Identifying Boundary Condition Detects network latency, system load; fallback with retries or messages. 				
	QR004	Interactive buttons should be responsive and execute actions immediately when tapped.	Performance	Response Time	Subsystem Venue Booking Subsystem CHAMP Point Management
	<p>Issues:</p> <ol style="list-style-type: none"> 1. Mapping Subsystem to Processors and Components Ensure the front-end is event-driven and responsive (e.g., React, Flutter). 2. Identifying and Storing Persistent Data Real-time interaction doesn't need persistent storage unless action is confirmed. 				

	<p>3. Providing Access Control Validate user rights before action execution.</p> <p>4. Designing the Global Control Flow Trigger UI updates immediately after confirmation from the backend.</p> <p>5. Identifying Boundary Condition Handle multiple rapid taps, failed button actions, or timeout errors.</p>				
Design Goal 3	QR005	The application should support an increasing number of users and events without requiring major architectural changes.	Scalability	System Scalability	Subsystem Venue Booking Subsystem CHAMP Point Management
<p>Issues:</p> <p>1. Mapping Subsystem to Processors and Components Design with microservices or scalable architecture; use cloud infrastructure.</p> <p>2. Identifying and Storing Persistent Data Use scalable databases (e.g., sharding, partitioning).</p> <p>3. Providing Access Control Should scale with a user base without bottleneck (e.g., stateless auth tokens).</p> <p>4. Designing the Global Control Flow Avoid centralized bottlenecks; use distributed message queues or event-driven flows.</p> <p>5. Identifying Boundary Condition</p>					

	Manage concurrent access, high load, and failover mechanisms.				
	QR006	The system should display real-time updates on venue and people availability.	Performance	Real-Time Feedback	Subsystem Venue Booking
	<p>Issues:</p> <ol style="list-style-type: none"> 1. Mapping Subsystem to Processors and Components Implement pub-sub or WebSocket mechanism for live updates. 2. Identifying and Storing Persistent Data Store latest status; use in-memory databases like Redis for real-time performance. 3. Providing Access Control Only authorized users should update/view real-time data. 4. Designing the Global Control Flow Push-based control to update clients dynamically. 5. Identifying Boundary Condition Handle user disconnects, partial updates, or stale data situations. 				
	QR007	Data related to events, committee members, and champ points should be accurate and consistent across all modules.	Data Quality	Data Consistency	Subsystem CHAMP Point Management
	<p>Issues:</p> <ol style="list-style-type: none"> 1. Mapping Subsystem to Processors and Components 				

	<p>Centralize data consistency logic in a shared service layer.</p> <p>2. Identifying and Storing Persistent Data Use a transactional database with integrity constraints and backup.</p> <p>3. Providing Access Control Restrict updates to authorized roles only; track changes with audit logs.</p> <p>4. Designing the Global Control Flow Ensure all modules use the same data source/API for accuracy.</p> <p>5. Identifying Boundary Condition Detects and recovers from sync issues between modules or data corruption.</p>
--	---

Table 8 : Table of Addressing Design Goals

4.0 DETAILED DESIGN

4.1 Navigation Design

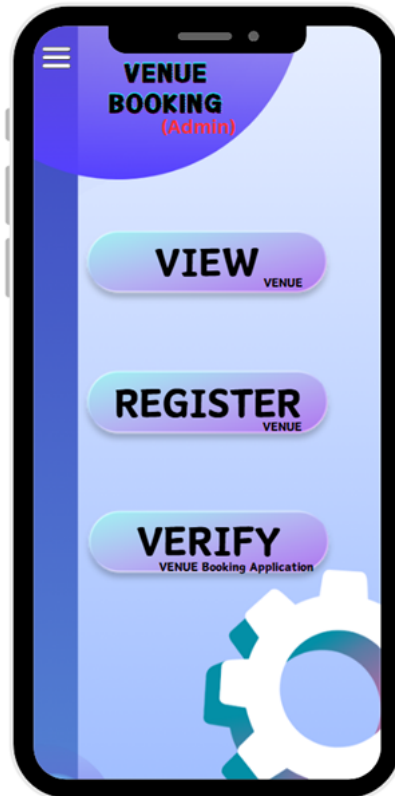
4.1.1 META Event Application Navigation Design

4.1.1.1 Subsystem Venue Booking (WONG CHUN LIANG)

MAIN MENU UI



VENUE BOOKING UI



Select Campus UI



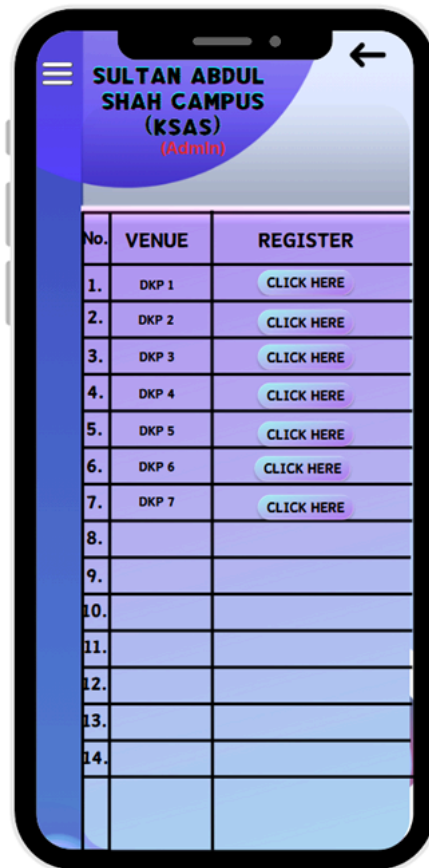
Show Venue UI In Campus KSAS (Subfunction View Venue)

No.	VENUE	STATUS	SIZE (SEAT)
1.	DKP 1	AVALABLE	100
2.	DKP 2	UNAVAILABLE	50
3.	DKP 3	UNAVAILABLE	100
4.	DKP 4	AVALABLE	100
5.	DKP 5	AVALABLE	200
6.	DKP 6	AVALABLE	100
7.	DKP 7	UNAVAILABLE	150
8.			
9.			
10.			
11.			
12.			
13.			
14.			

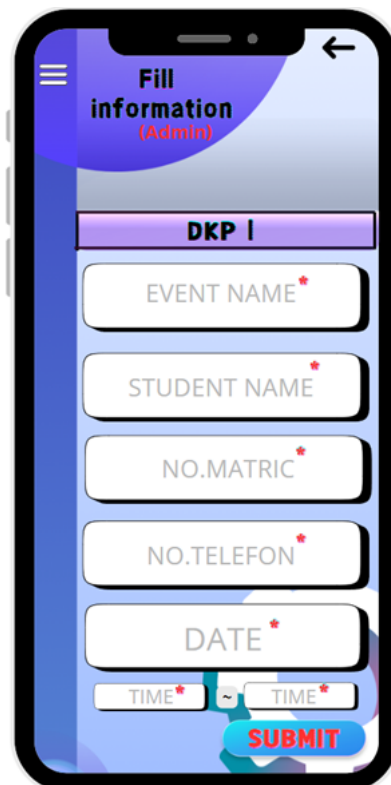
Show Venue UI In Campus KSAJS (Subfunction View Venue)

No.	VENUE	STATUS	SIZE (SEAT)
1.	DKA	UNAVAILABLE	100
2.	DKB	AVALABLE	50
3.	DKC	AVALABLE	100
4.	DKD	UNAVAILABLE	100
5.	DKE	AVALABLE	200
6.	DKF	AVALABLE	100
7.	DKG	UNAVAILABLE	150
8.			
9.			
10.			
11.			
12.			
13.			
14.			

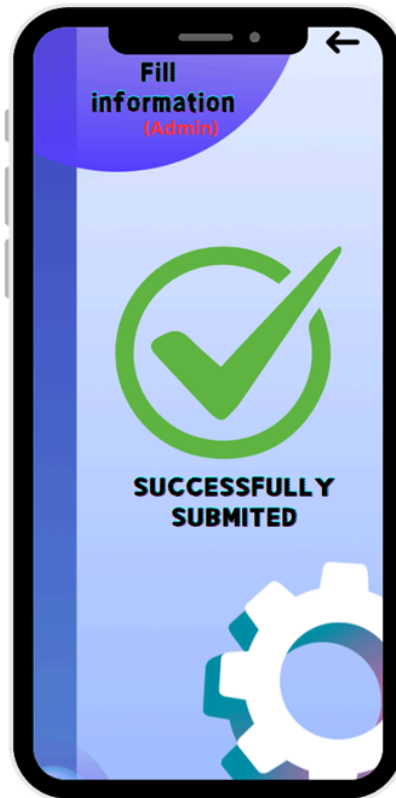
Select Venue UI (Subfunction Register Venue)



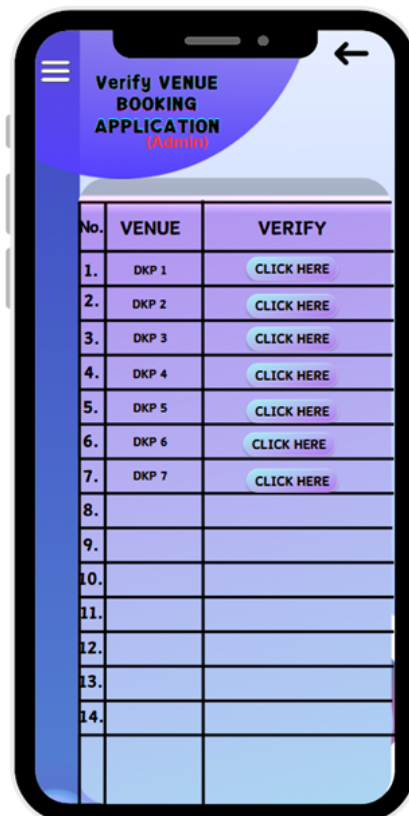
Fill Information UI (Subfunction Register Venue)



Show Successful Notice UI (Subfunction Register Venue)



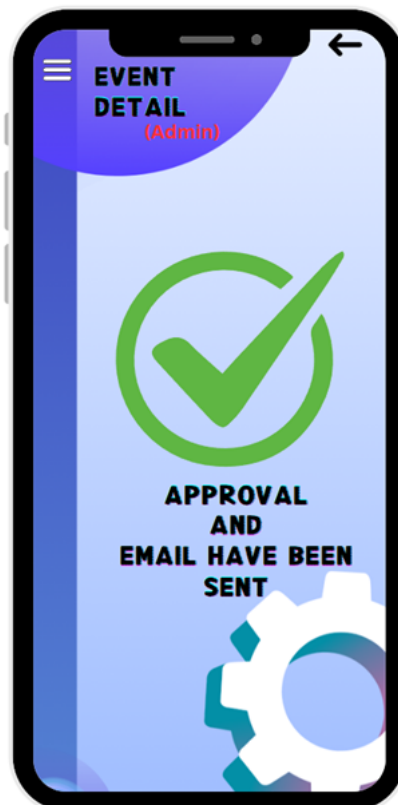
Select Venue UI (Subfunction Verify Application)



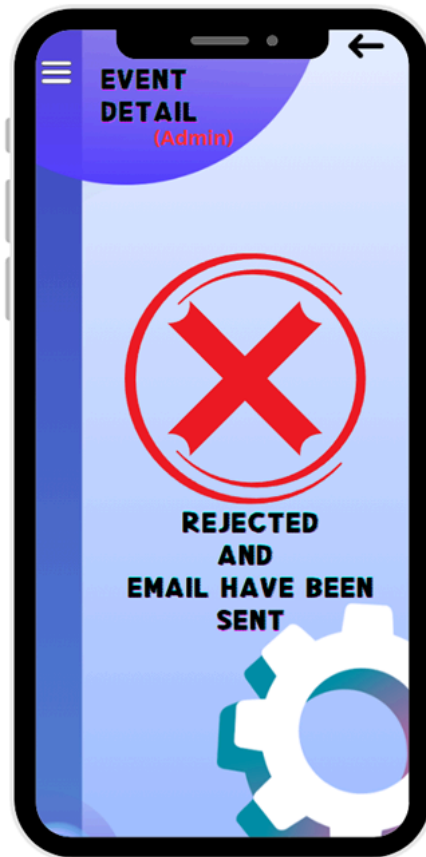
Show Event Detail UI (Subfunction Verify Application)



Show Notice UI (Subfunction Verify Application)



Show Notice UI (Subfunction Verify Application)

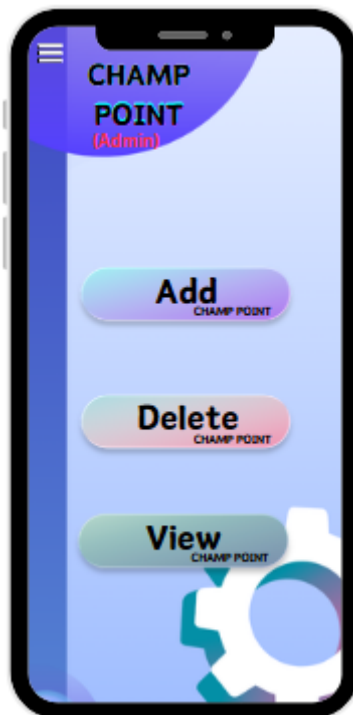


4.1.1.2 Subsystem CHAMP Point Management (GOH QUO TENG)

Main Menu



CHAMP Point Screen



Event List Screen (Subfunction Add)



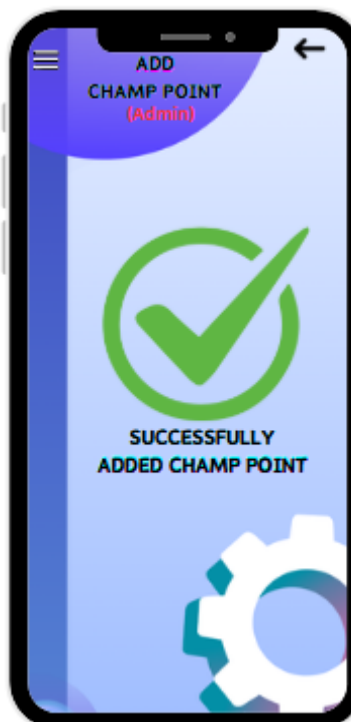
Participant List Screen (Subfunction Add)



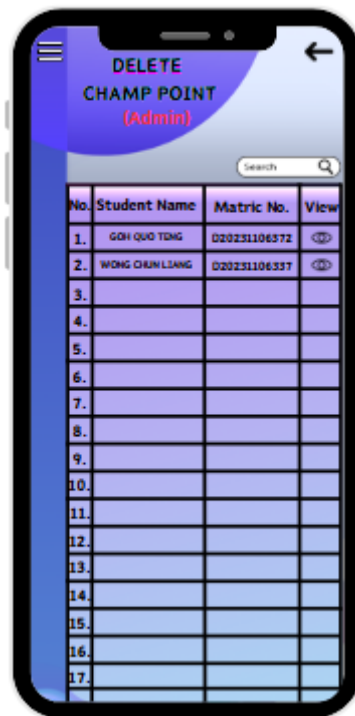
Add CHAMP Point Screen (Subfunction Add)



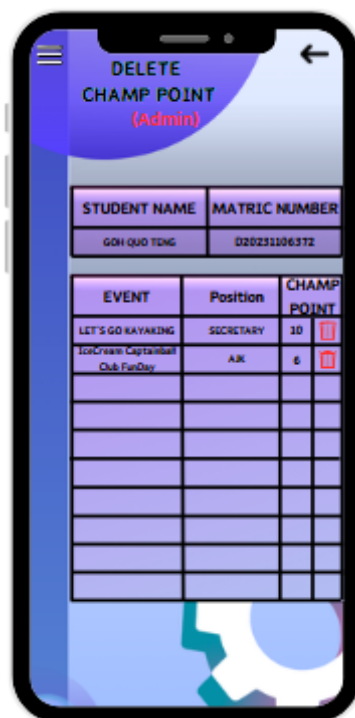
Successfully Added CHAMP Point Screen (Subfunction Add)



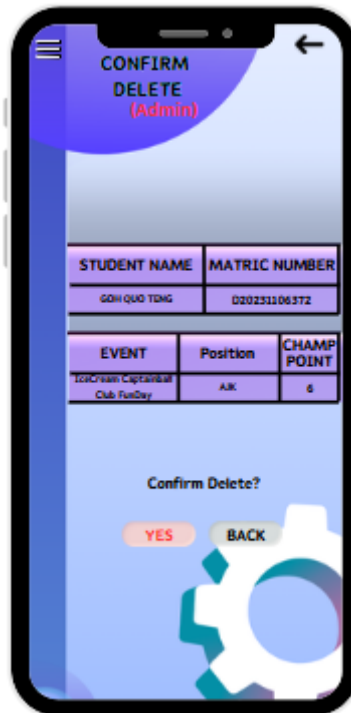
Participant List Screen (Subfunction Delete)



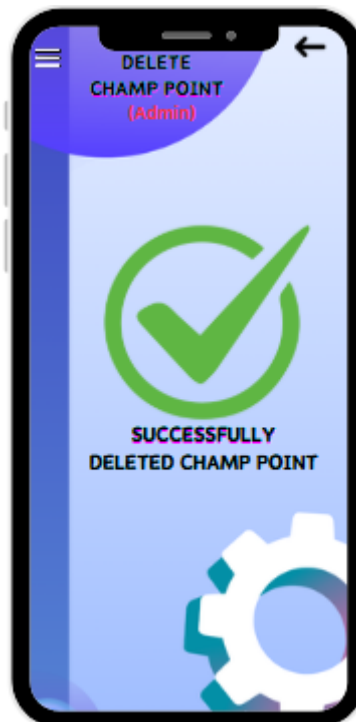
List Event That Student Participant in Screen (Subfunction Delete)



Delete CHAMP Point Screen Subfunction (Subfunction Delete)



Successfully Deleted CHAMP Point Screen (Subfunction Delete)



All Participants List Screen (Subfunction View)

No.	STUDENT	View
1.	GOH QUD TENG D20231106372	👁️
2.	WONG CHEN LIANG D20231106337	👁️
3.	PRISHA D20231106123	👁️
4.		
5.		
6.		
7.		
8.		
9.		
10.		
11.		
12.		
13.		
14.		
15.		
16.		
17.		
18.		

Total CHAMP Point Screen (Subfunction View)

STUDENT	
GOH QUD TENG	D20231106372

EVENT	POSITION	CHAMP POINT
LET'S GO KAYAKING	AJK	10
IceCream Captainal Club FunDay	PARTICIPANT	6

TOTAL	16
-------	----

4.2 Content Table

Subsystem	Importance Level	Content Name	Content to be written in the screen
Venue Booking Subsystem	High	View Venue	List of venues with details such as name, status and size.
		Register Venue	Fill in the following form to register a new venue.
		Verify Venue Booking Application	Review all pending venue booking applications.
CHAMP Point Management Subsystem	High	Add CHAMP Point	Fill in the points you want to add
		Delete CHAMP Point	Select the point you want to delete
		View CHAMP Point	Select student to view CHAMP point