1. Declaration

I, Ziqi Pei, declare that this assignment, titled Assignment3 Advanced Web App, is my own original work and has not been copied from any other source except where explicitly acknowledged. I have not engaged in plagiarism, collusion, or any other form of academic misconduct in the preparation and submission of this assignment. All sources of information and data used in this assignment have been properly cited and referenced in accordance with the prescribed guidelines. I have not used unauthorized assistance in the preparation of this assignment and have not allowed any other student to copy my work. I am aware that any breach of academic integrity may result in disciplinary action as per the policies of Monash University, which may include failing this assignment or the course, and further academic penalties.

Signature: _____Ziqi Pei_____ Date: _____13/10/2024_____

2. Github Check

Enter your Github details here.

Github Username *Enter your username here*	Franz-Pei/FIT5032
A3 Shared? *Have you started and shared your assignment repository with your tutor yet?*	https://github.com/Franz-Pei/FIT5032.git

3. Self-Evaluation

Criteria	Exceeds Expectations	Meets Expectations	Needs Improvement	Fail to meet expectations
BR (D.1): External Authentication	✅			
BR (D.2): Email	✅			
BR (C.3): Interactive Table Data	✅			
BR (D.4): Deployment to the Cloud	✅			
BR (E.1): Cloud Functions	✅			
BR (E.2): Geo Location	✅			
BR (E.3): Accessibility	✅			
BR (E.4): Export	✅			
BR (F.1): Innovation	✅			

4. Screen Recording of BRs

Create a 3 minute video showing your basic web application in action! Upload this video to your Google Drive and put the link here (ensuring that you have updated the access list so its not private).

https://drive.google.com/file/d/14qRR-Hy1N2hx5KahM8zrNIISz-7iAjA7/view?usp=sharing

5. BR F.1 - Innovative Features

1	Bulk Email	Bulk Email: This feature allows me to send emails to multiple users simultaneously, rather than individually. It's commonly used for marketing, notifications, or updates, saving significant time and effort. Selected Users: This functionality emphasizes not only bulk email sending but also the ability to target specific user groups. I can select users based on criteria such as user groups, subscription status, or other defined conditions, ensuring precise delivery to my audience. By linking Node.js with SendingGrade, the backend system can efficiently handle the bulk email sending process, ensuring smooth user selection, quick delivery, and reliable performance. In short, this feature will help me manage large-scale email campaigns efficiently, ensuring that messages are only sent to users meeting my specific criteria.
2	Appointment Booking (using Calendar):	sysFunction refers to the system feature used for managing appointment booking and handling constraints such as preventing booking conflicts. In your web application, you've implemented this functionality within the main.js file rather than using an external API like FullCalendar.io. The sysFunction in main.js ensures that no overlapping appointments are allowed by managing the logic for event scheduling and conflict detection directly within the core of the application. Additionally, this system is integrated with Firebase, meaning that all appointment data is saved and synchronized in real time. This ensures that updates to bookings are consistently stored and accessible across the application. In summary, the sysFunction in main.js manages appointment scheduling, conflict detection, and ensures that all booking data is

		securely saved to Firebase, enabling a smooth and efficient booking experience.
3	Provide API	SendEmailWithAttachment API (POST): This API allows external systems to send emails with attachments using SendGrid. The POST request includes parameters like recipient's email, subject, message, and attachments. The API returns the status of the email delivery. Endpoint: /sendEmailWithAttachmentAPI Request Body: to, subject, text, attachmentInfo Response: Success or failure status of the email delivery 1. Postman Test: Send a POST request with to, subject, text, and optionally attachmentInfo in base64 format. The response will indicate success or failure. 2. GetUserInfo API (GET): This API fetches user data from Firebase Storage. A GET request retrieves a JSON file containing all user information, which can be used by external platforms. Endpoint: /getUserInfoAPI Response: JSON file with user data from Firebase These APIs enable third-party platforms to integrate email services and access user data efficiently, automating workflows and facilitating data sharing between systems.
4	Gemini API	I've integrated GenAI (such as Gemini API or OpenAI API) into the platform, enabling advanced AI-driven capabilities. The GenAI models are used for various tasks, such as automating content generation, processing large datasets, or providing intelligent recommendations.The API interactions are seamlessly integrated into the existing system, allowing for dynamic responses and testing. Additionally, all relevant data generated or processed by the GenAI, such as responses or user-related information, is securely stored in Firebase for further use and real-time synchronization across the application. This ensures that both the AI capabilities

		and the backend data management are working together effectively. This approach enhances the functionality of the platform, providing smart, scalable AI features while maintaining secure and organized data storage.

6. Reflections: Challenges

What has been the most challenging part of this assignment for you? How has this stretched you as a programmer?

The most challenging part of this assignment has been managing the complexity and scope of integrating multiple advanced technologies while ensuring the project's objectives were met efficiently. Specifically, integrating tools like Firebase, SendGrid, Google Cloud, and Mapbox with GenAI has required me to step out of my comfort zone and adopt a more comprehensive, interconnected approach to development. Each tool presented unique challenges, especially with maintaining seamless communication between the services and ensuring security, scalability, and usability.

This assignment stretched me as a programmer in several key ways:

1. Handling API Integrations: Working with APIs like SendGrid and Firebase pushed me to understand RESTful service architecture and the importance of secure, reliable data transmission. Building and testing the APIs in Postman was essential for verifying the functionality and debugging the integration.
2. Applying GenAI for Practical Use Cases: Integrating GenAI for tasks like automating data model creation and optimizing database structures was new for me. Understanding the limits of automation and when to apply human intervention for business logic or security was a critical learning experience.
3. Complex Data Management: Managing diverse data types, including user appointments, emails, and geolocation data, forced me to rethink my approach to database design. Switching from local storage to Firebase Firestore required planning for cloud scalability and ensuring that the data architecture supported cross-platform use.
4. CI/CD Implementation: Setting up Continuous Integration/Continuous Deployment (CI/CD) pipelines for smooth development and deployment was an essential learning curve. Automating testing and deployment processes ensured the stability of the system and improved my workflow efficiency.
5. Security and Privacy: As the project involved handling sensitive user data, it was a challenge to implement proper security measures. This stretched my understanding of data encryption, API authentication, and access control.

These challenges have helped me grow not only in terms of technical skills but also in problem-solving and decision-making, particularly when balancing automation with manual intervention for more complex features.

7. Declaration: Additional Help

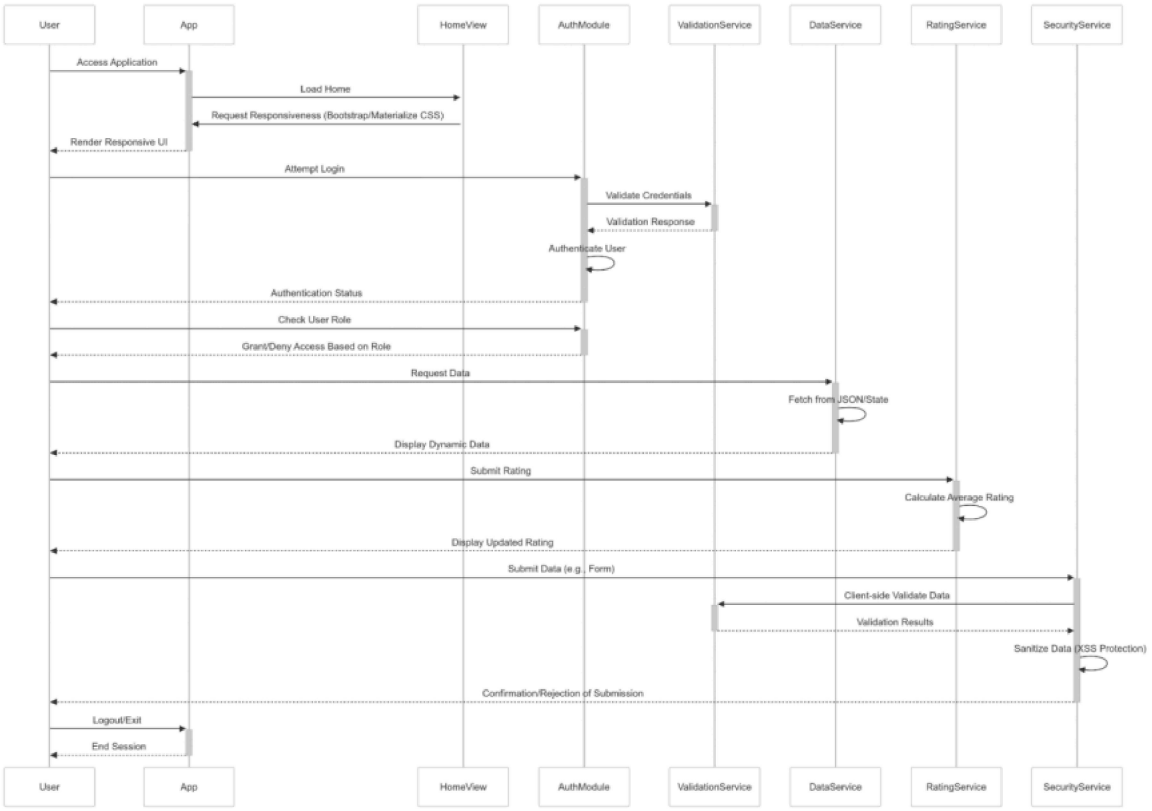
Name	Description
ChatGPT	I used ChatGPT to brainstorm the structure of APIs, discuss the pros and cons of using Firebase for data storage, and solve integration challenges with SendGrid. I also used it for brainstorming problem-solving strategies when working with Firebase Functions.
mermaid	Used Mermaid to generate Entity Relationship Diagrams (ERDs) automatically, helping visualize the data structure.
Firebase Authentication	Implemented user authentication on the platform to ensure secure login and registration functionalities.
Firestore Database	Used Firestore as the primary database for storing user data and email records, facilitating real-time access and scalable data management.
FirebaseFunction	Created cloud functions to handle email-sending logic using SendGrid, ensuring that the backend could handle server-side operations efficiently.
Firebase Storage	Used Firebase Storage to manage and store user-uploaded files, ensuring secure and scalable cloud storage solutions.
LucidChat	Utilized LucidChart to design flow diagrams for better planning and visualization of user flows and data structures.

Sendgrid	Integrated SendGrid API for sending bulk emails to users, including attachment support, to enhance communication capabilities.
Mockaroo	Used Mockaroo to generate mock data for testing interactive data tables and APIs before actual deployment.
Gemini API Vertex AI	Incorporated Gemini API Vertex AI to configure user interaction feedback on the web application
Google Drive Video	Hosted and shared video demonstrations of the application and its features for user testing and feedback purposes.
Cloudflare	Deployed the platform using Cloudflare Pages to ensure secure and fast delivery of web content to users.
Mapbox	Integrated Mapbox API for geolocation and address search functionality, including storing search history in Firestore for future reference.
Accessibility Checker	Used accessibility checker tools to ensure the web application met WCAG 2.1 AA standards for users with disabilities.
Syncfusion Calender	Integrated Syncfusion Calendar for managing and displaying user appointments with added functionality for handling booking constraints.
Google Cloud log Explorer	Utilized Google Cloud Log Explorer to monitor Firebase Functions and ensure proper logging for debugging and performance monitoring.
Github	Used GitHub for version control and collaboration during the development process, ensuring seamless code management and CI/CD integration.

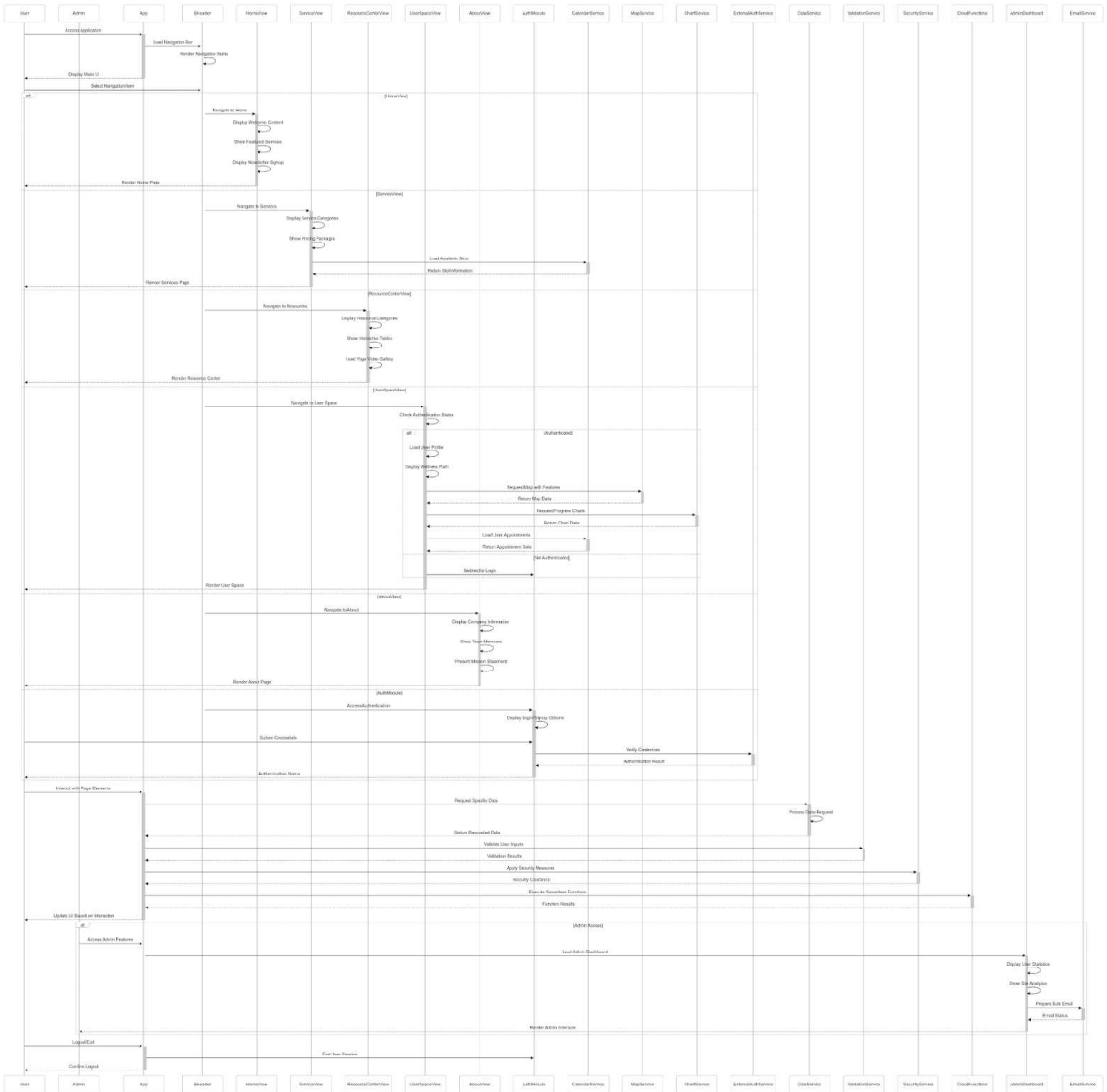
Trello board link

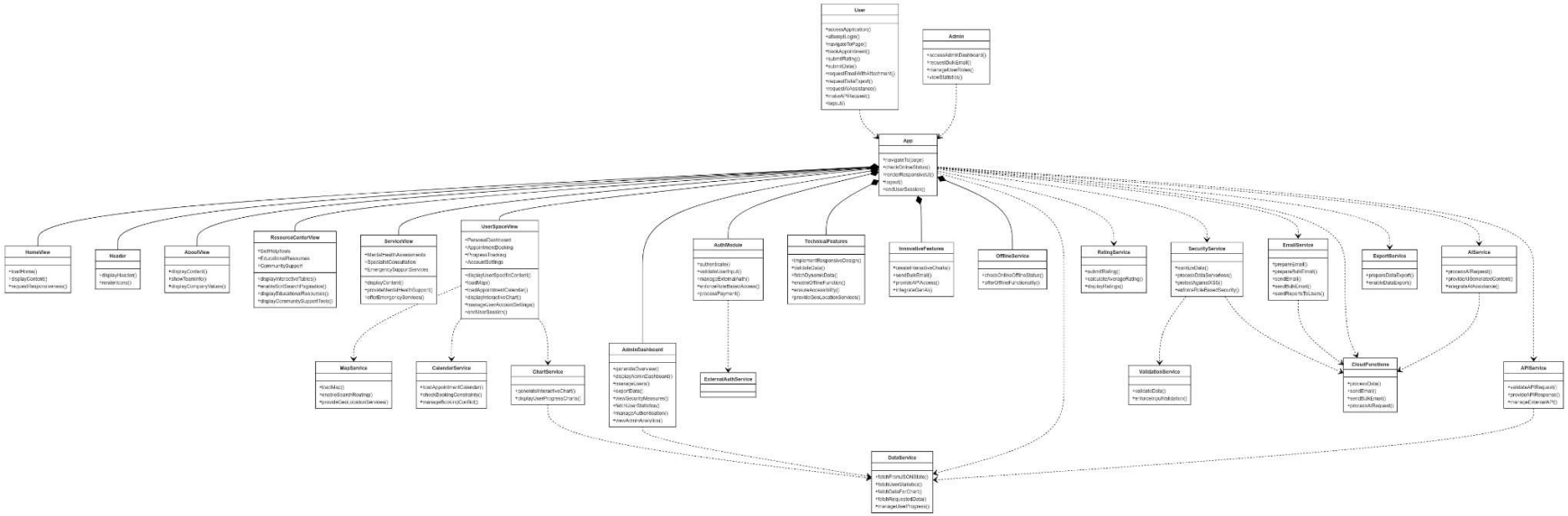
https://trello.com/invite/b/66fbb4949d55f80e16e8f04b/ATTIfda43780534da9718e3d41a4c97aa0aaCECA7E51/ziqi-peis-fit5032-assignment-d-e

Past time Sequence Diagram



Update Seuquence Diagram




User
Admin
App
HomeView
Header
AboutView
ResourceCenterView
ServiceView
UserSpaceView
AuthModule
TechnicalFeatures
InnovativeFeatures
OfflineService
RatingService
SecurityService
EmailService
ExportService
AIService
MapService
CalendarService
ChartService
AdminDashboard
ExternalAuthService
ValidationService
CloudFunctions
APIService
DataService