# 

Adama Science And Technology University

School of Electrical Engineering and Computing

Software Engineering Department

Course Title: Mobile Application Design And Development

Course Code: SEng3206

Project Title: Campus Bites

Group Name ID

1. Chala Temesgen ………………………… ugr/30356/15

2. Dawit Worku ……………………………… ugr/30395/15

3. Hamza kedir ……………………………… ugr/30618/15

4. Nezira Worku …………………………….. ugr/31080/15

5. Tomas Melesse ………………………….. ugr/31318/15

6. Tsion Wubshet …………………………… ugr/31333/15

7. Wubgzer Alemayehu ……………………. ugr/31349/15

Submitted to: Mr. Yared Tekaling

Date of Submission: May 2, 2025

**Table of contents**

Catalog

[Chapter 1. Introduction Project Overview 1](#_Toc5966)

[1.1 Background 1](#_Toc12859)

[1.2 Problem Statement 1](#_Toc12883)

[1.3 Project Objectives 2](#_Toc8163)

[1.3.1 General Objective 2](#_Toc11354)

[1.3.2 Specific Objectives: 2](#_Toc27500)

[1.4 Scope and Limitations 3](#_Toc5377)

[1.4.1 scope 3](#_Toc26451)

[1.4.2 Limitation 4](#_Toc14553)

[Chapter 2. User Requirements 4](#_Toc2501)

[2.1 Funcitional Requirement 4](#_Toc10676)

[Chapter 3. Design Concepts 7](#_Toc10303)

[Chapter 4. Development Approach 9](#_Toc3657)

[4.1 Methodology Selection 9](#_Toc285)

[4.2 Challenges and Solutions 9](#_Toc7160)

[Chapter 5. Technological Stack 10](#_Toc2769)

[5.1 Frontend Technologies 10](#_Toc29625)

[5.2 Backend Technologies 11](#_Toc32148)

[5.3 Development Tools and Environment 12](#_Toc9085)

[Chapter 6. Implementation Details 12](#_Toc29640)

[6.1. User Authentication 12](#_Toc22401)

[6.2 Menu Browsing and Item Details 13](#_Toc7798)

[6.3 Profile Management 14](#_Toc25452)

[Chapter 7. Testing and Quality Assurance 16](#_Toc13945)

[7.1 Testing Strategies 16](#_Toc18736)

[7.2 Effectiveness and Reflection 16](#_Toc22626)

[Chapter 8. Future Enhancements 17](#_Toc21499)

8.1 In-App Payment Integration ............................................................................17

8.2 Real-Time Delivery Tracking ............................................................................17

[8.3 Loyalty Programs and Discounts 18](#_Toc10260)

[8.4 Community Reviews and Ratings 18](#_Toc20422)

[Refereces 19](#_Toc17067)

# 

# Chapter 1. **Introduction Project Overview**

* 1. **Background**

Modern university campuses are bustling ecosystems where students, faculty,and staff juggle academic, professional, and personal responsibilities. Access to convenient, diverse, and affordable food options is critical to sustaining productivity and well-being. However, traditional campus cafeterias often fall short due to limited operating hours, repetitive menus, and long queues during peak times. Off-campus food delivery platforms (e.g., Hey Eats, Yummy Delivery ) are rarely optimized for campus communities, leading to delayed deliveries, high fees, and fragmented vendor partnerships.

Campus Bites was created by our team in 2025 to solve everyday problems faced by students and staff on university campuses. Many people struggle to find quick, affordable meals between classes or during late-night study sessions. Campus cafeterias often close early, have long lines, or lack menu variety, leaving users frustrated. At the same time, local restaurants and hotels near campuses want to serve the student community but find it hard to manage orders manually or rely on expensive third-party delivery apps that charge high fees. Our team saw an opportunity to bridge this gap by building a dedicated platform for campuses—connecting hungry users directly with nearby vendors in a way that’s faster, cheaper, and simpler for everyone.

* 1. **Problem Statement**

The current food ecosystem on university campuses suffers from inefficiencies that negatively impact students, staff, and vendors. Campus cafeterias often operate on limited schedules, closing during evenings , leaving users with no affordable or convenient meal options. Long queues during peak hours further exacerbate delays, forcing students to miss meals or resort to unhealthy snacks. Meanwhile, local hotels and restaurants near campuses struggle to tap into the student market due to fragmented ordering systems, reliance on third-party delivery apps with high commission fees, and manual order management processes that lead to errors. Students and staff also face challenges tracking deliveries in real-time, resulting in missed meals or cold food, while vendors lack tools to forecast demand, leading to food waste or stock shortages.

* 1. **Project Objectives**
     1. **General Objective**

The primary goal of Campus Bites is to develop a centralized, campus-specific food ordering and delivery platform that bridges students, staff, and local vendors. The system aims to eliminate inefficiencies in traditional campus food services by providing a seamless digital experience, reducing dependency on third-party apps, and promoting sustainable practices. By integrating real-time order management, dynamic vendor partnerships, and user-friendly features, Campus Bites strives to enhance accessibility, affordability, and convenience for all stakeholders while fostering a more connected and Eco-conscious campus community.

The platform seeks to empower vendors with tools to streamline operations, minimize waste, and grow their customer base, while ensuring users gain access to diverse, affordable meals with minimal wait times. Simultaneously, Campus Bites aligns with institutional goals by reducing campus carbon footprints, supporting local businesses, and creating a salable model adaptable to diverse academic environments.

* + 1. **Specific Objectives:**
* User-Centric Ordering System

Develop a mobile and web interface for students/staff to:

* + - Register accounts using campus credentials (e.g., university email).
    - Browse real-time menus from partnered vendors with filters for dietary needs (e.g., vegan, halal).
    - Place orders, track delivery status via GPS, and rate services.
* Vendor Management Tools

Provide vendors with a dashboard to:

* + - Manage menus (add/remove items, update prices).
    - Accept/reject orders, estimate preparation times, and assign delivery agents.
    - Analyze sales trends and forecast demand to reduce food waste.
* Sustainability Initiatives
  + Implement AI-driven demand forecasting to help vendors optimize ingredient purchases.
  + Incentive eco-friendly packaging by highlighting “green vendors” in the app and offering discounts for reusable containers.
* Administrative Oversight
  + Equip campus administrators with a dashboard to:
    - Approve/remove vendors, monitor hygiene ratings, and resolve disputes.
    - Generate reports on order volumes, revenue trends, and user feedback.
* Accessibility and Inclusive
  + Ensure the platform complies with accessibility standards (e.g., screen readers for visually impaired users).
  + Provide multilingual support and prioritize vendors offering affordable meal plans for low-income students.
* Performance Optimization
  + Design the system to handle concurrent users during peak hours with few second response times.
  + Ensure 99.9% uptime and robust data encryption for secure transactions.
* Partnership Development
  + Establish agreements with local vendors within the first year of launch.
  + Collaborate with campus sustainability clubs to promote zero-waste initiatives.
  1. **Scope and Limitations**

1. **scope**

* Core Functionality Campus Bites will focus on developing a web and mobile platform for students, staff, and vendors to manage food ordering and delivery. Key features include user registration, real-time menu browsing, order placement, payment processing, delivery tracking,and vendor dashboards. The system will integrate with campus authentication systems (e.g., university email) and payment gateways (e.g., meal plans). Administrative tools for user management, vendor approvals, and reporting are also included.
* Vendor Partnerships The platform will onboard hotels, restaurants, and cafes within a 3-mile radius of the campus to ensure timely delivery. Vendors will gain access to a dashboard to update menus, manage orders, and view analytic . Partnerships will prioritize vendors offering student discounts or eco-friendly packaging.

1. **Limitation**

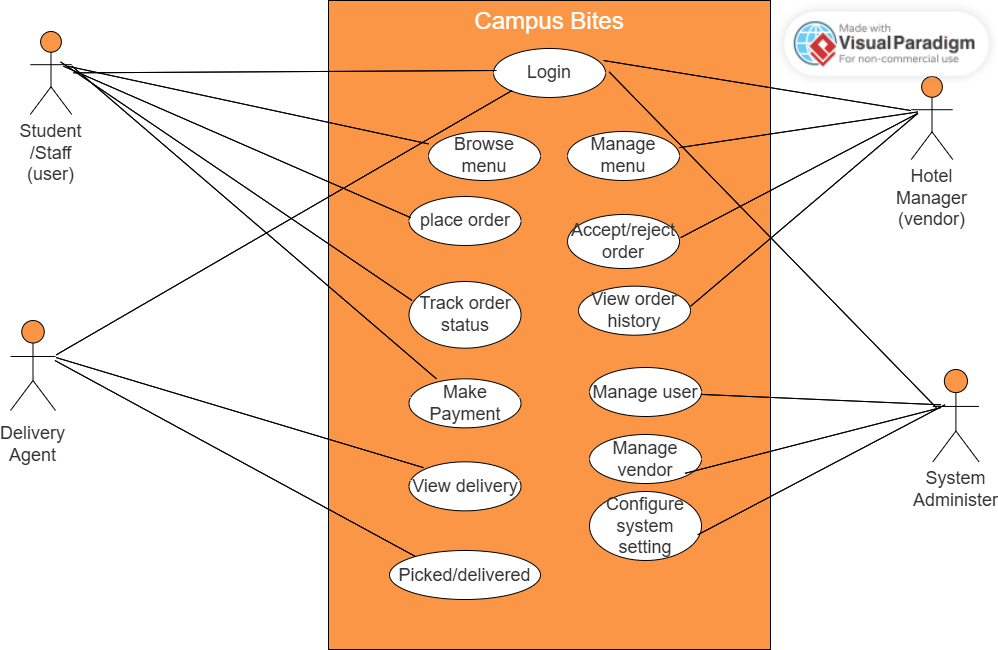
* Geographic and Vendor Constraints Campus Bites will only serve users within the campus boundaries and its immediate vicinity (3-mile radius). Delivery to off-campus locations (e.g., student housing outside this range) is excluded. Additionally, vendors must agree to campus-specific terms (e.g., discounted pricing) to join the platform, limiting participation to those willing to comply.
* Excluded Features The system will not manage food preparation logistics, inventory, or delivery personnel hiring for vendors. Delivery coordination (e.g., driver assignments) remains the vendor’s responsibility.

# Chapter 2. **User Requirements**

1. **Funcitional Requirement**

* four key use case diagrams for Campus Bites:

1. User (Student/Staff) Interactions
   * Actors: Student, Staff
   * Use Cases:
     + Register/Login
     + Browse Menu
     + Place Order
     + Track Delivery
     + Make payment
   * Diagram Focus: Shows how users interact with the platform to order food and manage their accounts.
2. Vendor (Hotel Manager) Interactions
   * Actors: Hotel Manager
   * Use Cases:
     + Login
     + Manage Menu (Add/Remove Items)
     + Accept/Reject Orders
     + View order history
   * Diagram Focus: Illustrates how vendors manage their menus and order.
3. Delivery Agent Interactions
   * Actors: Delivery Agent
   * Use Cases:
     + Login
     + View Assigned Orders
     + Confirm Delivery Completion
   * Diagram Focus:Depicts delivery workflows from pickup to drop-off.
4. Admin Interactions
   * Actors: System Admin
   * Use Cases:
     + Login
     + Manage Users (Approve/Block Accounts)
     + Manage Vendors (Approve/Remove Partners)
     + Configure system setting
   * Diagram Focus: Highlights administrative oversight and system management.

figure 2.1 use case diagram

* 1. **Non-Functional Requirements**

Campus Bites must ensure reliability by maintaining system uptime to prevent disruptions during peak ordering times, such as lunch hours or late-night study sessions. The platform should handle unexpected failures gracefully, such as payment processing errors, by automatically retrying transactions or switching to backup systems without data loss. Security is critical, requiring end-to-end encryption for all transactions, secure authentication (e.g. login).

Usability is prioritized through an intuitive, mobile-first interface with accessibility features like screen reader compatibility and adjustable font sizes. Scalability is essential to accommodate growing user bases, allowing the system to scale horizontally during high-traffic events without performance degradation.

Performance demands include response times for all user actions, such as loading menus or processing payments. The system must efficiently manage background tasks, such as real-time delivery tracking and push notifications, without slowing down core functionalities. Maintainability is ensured through modular code architecture, comprehensive documentation, and automated testing frameworks, allowing for seamless updates and bug fixes.

# Chapter 3. **Design Concepts**

The design of the **Campus Bites Mobile App** emphasizes **simplicity and clarity**, guided by modern **Material Design principles**. We adopted a clean, minimalist UI that focuses on essential elements to reduce cognitive load for users. Icons, color schemes, and fonts were chosen to reflect a youthful, campus-friendly vibe while maintaining professionalism and readability. Consistent spacing, alignment, and a unified color palette contribute to a visually appealing and intuitive interface.

From a **user experience (UX)** standpoint, the app prioritizes **efficiency and ease of use**. Key actions—like browsing menus, adding items to the cart, and checking out—are reachable within two taps. We implemented a responsive layout and optimized user flows to minimize friction during ordering. First-time users are guided with brief tool tips, and all interactions provide clear feedback (like loading indicators, success/error messages, and animations) to improve usability and trust.

In terms of **navigation flow**, we use a **bottom navigation bar** to streamline access to core sections: Home, Orders, Cart, and Profile. This layout aligns with common mobile UX patterns, reducing the learning curve. Visual elements such as high-quality food images, custom icons, and smooth transitions were integrated to enhance the aesthetic experience and engagement, while maintaining accessibility standards for a diverse user base.

Accessibility was a key consideration throughout the design process. The app supports saleable text, clear contrast ratios, and touch-friendly elements to ensure that users with visual or motor impairments can navigate and use the app comfortably. We've also ensured that error states, such as failed payments or invalid form inputs, are communicated with clear, descriptive messages and suggestions for resolution, helping users recover quickly without frustration.

We also placed strong emphasis on **consistency and feedback**. Buttons, input fields, and other interactive elements follow a uniform style across all screens, making the experience predictable and easy to follow. Micro-interactions—like subtle vibrations on button taps, loading animations, and progress bars—help reinforce a sense of control and responsiveness. Together, these design decisions create a smooth, engaging, and user-centered experience tailored for campus environments.

To create a vibrant and welcoming feel, we carefully selected a color palette dominated by warm tones like orange and soft green, which psychologically stimulate appetite and positive. We used high-resolution images for food items and intuitive iconography to make browsing menus more enjoyable. Typography also plays an important role; we combined bold headings for clear navigation with lighter text for secondary information, ensuring that users can easily distinguish primary actions from supplementary details.

# Chapter 4. **Development Approach**

* 1. **Methodology Selection**

For the development of the **Campus Bites Mobile App**, we adopted the **Agile methodology**, specifically a Scrum-inspired approach. Agile was chosen because it allows for rapid iteration, continuous feedback, and the flexibility to adapt to changing requirements. Since food delivery services often need to respond to real-world feedback from users and vendors, it was important for us to maintain a dynamic workflow. By breaking down the project into sprints, we could focus on delivering key features incrementally, starting with user authentication and menu browsing before moving on to order tracking and payment integration.

Moreover, Agile enables **collaborative teamwork**, which was essential given that the project involved cross-functional collaboration between developers, UI/UX designers, and back-end engineers. Regular sprint meetings, including daily stand-ups and sprint retrospectives, ensured that everyone stayed aligned and any potential roadblocks were addressed promptly. This transparent and flexible approach allowed us to make adjustments during the development process based on evolving user requirements and real-time feedback, ensuring the app would meet the needs of both the users and the campus environment.

Additionally, we chose Agile to ensure continuous **product improvement**. As users interacted with the app, new features or optimizations often became apparent, and the iterative nature of Agile allowed us to prioritize and implement these enhancements quickly. This was particularly helpful in creating an evolving mobile app where updates and changes could be rolled out seamlessly without disrupting the user experience. Through Agile, we were able to stay aligned with our core goals while adapting to the ever-changing requirements of the campus food ordering and delivery market.

* 1. **Challenges and Solutions**

During the development process, we encountered several challenges. One of the biggest hurdles was ensuring real-time updates for order statuses without draining device performance. To address this, we implemented efficient background polling mechanisms and optimized API calls to limit unnecessary network usage. Another challenge was maintaining synchronization between the mobile app and the back end, especially when handling high volumes of simultaneous orders. We overcame this by designing a robust API contract early on and regularly updating both front end and back end teams through sprint review meetings, ensuring smooth communication and consistent updates.

Additionally, integrating user feedback into the development cycle was initially challenging due to scheduling constraints among testers and stakeholders. To solve this, we created lightweight feedback forms and organized short, focused testing sessions at the end of each sprint. This allowed us to quickly gather actionable insights without slowing down the development pace. Overall, the Agile approach proved highly effective in helping us build a user-friendly, reliable app that could evolve alongside the needs of its campus community.

Another major challenge we faced was ensuring a consistent and intuitive user experience across different device types and screen sizes. Since our target users have a wide range of smartphones with varying performance levels, screen resolutions, and operating systems, we needed to make sure the app was responsive and functional everywhere. To address this, we conducted extensive testing on both Android and iOS devices, using emulators and real devices. We also adopted adaptive layout techniques and optimized our assets and code to maintain smooth performance without sacrificing the visual quality of the app.

# Chapter 5. **Technological Stack**

* 1. **Frontend Technologies**

For the development of the Campus Bites mobile application, we selected **React Native** as our primary front end framework. React Native allows us to build high-performance, cross-platform applications from a single code base, significantly reducing development time and maintenance costs. Since we wanted to deploy the app on both Android and iOS platforms simultaneously, React Native provided the perfect balance between native performance and development efficiency.

We enhanced the structure and safety of our code base by using **Type Script**. Type Script’s strong typing system helped catch errors early during development and made the code base more saleable and maintainable. With a growing app like Campus Bites that includes

features like real-time order tracking and secure authentication, Type Script ensured better documentation, improved developer experience, and easier collaboration among team members.

For styling, we integrated **Tailwind CSS** with React Native using libraries like tailwind-Rn. Tailwind allowed us to apply consistent and reusable utility-first styles rapidly. This approach not only sped up UI development but also maintained a clean and organized design system. The combination of Tailwind and React Native enabled us to create a visually appealing, responsive, and efficient user interface aligned with modern mobile design standards.

* 1. **Backend Technologies**

On the back end, we chose **App write** as our back-end-as-a-service (Baas) solution. App write provided a comprehensive suite of tools including authentication, databases, cloud functions, and storage — all in a single, self-hosted platform. This allowed us to focus more on app features rather than building back end infrastructure from scratch, saving time and ensuring reliability from the early stages of development.

Using App write is secure authentication system, we were able to implement email/password registration and login functionalities efficiently. Additionally, App write is real-time database and server less functions gave us the flexibility to create dynamic features such as live order updates and notifications without building complex back end services. This significantly accelerated our ability to launch key features while maintaining strong data security and performance.

Moreover, App write is compatibility with modern front end technologies like React Native made integration straightforward. The REST and Graph QL API provided by App write enabled seamless communication between the app and the back end. Its scalability and modular architecture also ensured that as Campus Bites grows in users and features, our back end will be capable of handling the increasing load without major refactoring efforts.

* 1. **Development Tools and Environment**

To support our development workflow, we used **Visual Studio Code (VS Code)** as the primary code editor, benefiting from its wide range of extensions tailored for React Native and Type Script development. Tools like ES Lint and Prettier were also integrated into the project to enforce coding standards and maintain clean, readable code across the entire team.

We used **Git** and **GitHub** for version control and project collaboration. GitHub allowed multiple developers to work simultaneously through branches and pull requests, ensuring that new features and bug fixes were integrated carefully without disrupting the main production code. Continuous integration workflows were set up using GitHub Actions to automate testing and builds, further enhancing code quality and deployment speed.

For testing and debugging, we leveraged tools like **React Native Debugger**, **Jest** for unit testing, and **Expo** during early development stages for faster prototyping. These tools significantly improved our development efficiency, helping us identify and resolve issues quickly, and ensuring the final product delivered a stable and polished experience to end-users.

# Chapter 6. **Implementation Details**

1. **User Authentication**

The Campus Bites application provides a secure and seamless **user authentication** system. Users can sign up using their email and password, and log in securely with App write authentication service. We implemented form validations to ensure strong password policies and correct email formats, enhancing security and user trust. After logging in, users are redirected to the home screen, where they can start browsing food options immediately.

We also implemented features like password reset and session management to improve the user experience. Users who forget their passwords can easily recover access via a reset email link. Furthermore, secure session handling ensures that user data is protected while using the app.

|  |  |  |
| --- | --- | --- |
| page1 | page2 | page3 |

Figure 6.1 screen shoots of User Authentication

1. **Menu Browsing and Item Details**

A core feature of the app is the **menu browsing experience**, where users can explore available dishes from different partner hotels. We implemented dynamic menu loading using Appwrite’s real-time database. The design focuses on clean presentation: food images, dish names, descriptions, and prices are displayed clearly to make selection easy and engaging.

When a user taps on a menu item, they are taken to an **Item Detail screen** showing a larger image, full description, ingredients, and an "Add to Cart" button. This helps users make informed decisions before placing an order.

|  |  |  |
| --- | --- | --- |
| **home** | **hom2** | **home3** |

Figure 6.2 screen shoots of Menu Browsing and Item Details

1. **Profile Management**

The **Profile** section allows users to update their personal information, including name,phone number, and password. We implemented secure form handling with real-time validation to make updates simple and safe.

Users can also log out from their profile page with a single click, which securely ends their session and redirects them to the login screen, maintaining user privacy.

Additionally, the profile area includes important features such as account deletion and settings customization. Users who wish to permanently delete their account can do so securely after confirmation prompts to prevent accidental deletions. We also introduced a dark mode option within settings, allowing users to switch between light and dark themes based on their preference. These features enhance user control over their experience and cater to person needs.

|  |  |
| --- | --- |
| setting | setting2 |

Figure 6.3 screen shoots of Profile Management

The above all screenshots showcase some of the key user interfaces of the Campus Bites mobile application, including login and registration screens, menu browsing, item details, cart management, order tracking, and profile settings. These visuals highlight the clean design, intuitive navigation, and user-centered features that enhance the overall food ordering experience on campus.

# Chapter 7. **Testing and Quality Assurance**

1. **Testing Strategies**

As university students, we employed a combination of **manual testing** and **user acceptance testing (UAT)** to ensure the functionality, performance, and reliability of the Campus Bites mobile application. Since we had limited access to professional testing resources, manual testing allowed us to systematically check each feature — including registration, login, browsing menus, placing orders, and tracking deliveries — across different devices and network conditions. We prepared a checklist of core functionalities and tested them after each sprint to detect bugs early and ensure that every feature worked as expected.

In addition to feature testing, we conducted **device compatibility testing** by running the app on various Android and iOS devices available among our classmates. This helped us identify UI issues on different screen sizes and hardware capabilities. To simulate real-world campus scenarios, we tested the app under unstable Wi-Fi and mobile data connections to assess how well the app handled network errors and offline scenarios. Overall, these hands-on strategies were effective for a university-level project and allowed us to deliver a stable application without the need for expensive automated testing tools.

1. **Effectiveness and Reflection**

Our testing approach was largely effective in ensuring the application met basic functionality, user expectations, and reliability standards. By conducting iterative manual testing after each development sprint, we were able to identify and fix small issues quickly, preventing them from growing into larger problems later. Having our fellow students act as testers during the UAT phase was especially helpful, as they represented real users who would be using the app on campus. Their feedback on user experience, performance, and design helped us refine the app significantly before final submission.

However, we also recognized some limitations in our testing process. Without automated testing tools, some repetitive tasks like regression testing were time-consuming and prone to human error. Additionally, due to limited device access, we were unable to test on every possible screen resolution or operating system version. In the future, to improve the quality assurance process for larger-scale projects, we plan to integrate unit testing frameworks like **Jest** for front end logic and use mobile testing services like **Browser Stack** for better device coverage. Despite these limitations, our testing methods were appropriate for our university environment and contributed to a functional and reliable final product.

# Chapter 8. **Future Enhancements**

1. **In-App Payment Integration**

One of the major enhancements planned for Campus Bites is the **integration of in-app payment systems**. Currently, payments are handled manually upon delivery or through external mobile banking applications. Introducing secure in-app payment options like credit/debit card processing, mobile wallets (e.g., Telebirr, CBE Birr), and PayPal would significantly streamline the ordering process. Users would be able to complete transactions within the app, improving convenience and reducing delays during the delivery process.

Integrating payments directly into the app would also align perfectly with our project's objective of providing a **seamless and hassle-free campus food ordering experience**. Additionally, it would build trust and loyalty among users who prefer cashless transactions. Security measures such as SSL encryption and payment gateway integration would ensure that user financial information remains protected.

1. **Real-Time Delivery Tracking**

Another important enhancement is the implementation of **real-time GPS-based delivery tracking**. Although the current version updates order status manually ("Preparing," "On the way," etc.), real-time tracking would allow users to see the live location of their delivery on a map. This would greatly enhance transparency and user satisfaction by reducing uncertainty about order arrival times.

Adding this feature would align with modern user expectations, particularly for a student population that is accustomed to apps like Uber Eats and Glovo. It would also help delivery drivers by optimizing routes and improving delivery efficiency, benefiting both the users and hotel partners collaborating with Campus Bites.

1. **Loyalty Programs and Discounts**

To encourage frequent use of the application, we propose introducing a **loyalty program** and **discount system**. Users could earn points for every order placed, which could later be redeemed for discounts, free items, or exclusive promotions. Personalized discount notifications based on user preferences could also be implemented to drive engagement.

These features would not only enhance user retention but also support our project’s goal of building a vibrant, ongoing relationship between students and local food vendors. By incentive's repeat orders, both users and hotels benefit, ensuring the platform's long-term success on campus.

1. **Community Reviews and Ratings**

Adding a **review and rating system** for hotels and menu items would empower users to share their dining experiences. Users could leave comments, rate food quality, delivery speed, and customer service, helping others make informed decisions before ordering.

Implementing this feature would further the Campus Bites mission of transparency and quality assurance. It would also create healthy competition among hotel partners, encouraging them to maintain high service standards to earn better ratings and more orders from students.

**Refereces**

1. React Native Documentation — [https://reactnative.dev/docs/getting-started](https://reactnative.dev/docs/getting-started )
2. TypeScript Handbook — Microsoft Docs — [https://www.typescriptlang.org/docs/](2. https:/www.typescriptlang.org/docs/)
3. Tailwind CSS Documentation — [https://tailwindcss.com/docs](https://tailwindcss.com/docs" \t "_new)
4. Appwrite Documentation —  [https://appwrite.io/docs]( https:/appwrite.io/docs)
5. Krug, S. (2014). Don't Make Me Think: A Common Sense Approach to Web Usability. 3rd Edition, New Riders Publishing.
6. Nielsen, J. (1994). Usability Engineering. Morgan Kaufmann.
7. Pressman, R. S. (2014). Software Engineering: A Practitioner's Approach. 8th Edition, McGowan-Hill Education.
8. Google Material Design Guidelines — <https://m3.material.io/>
9. Agile Alliance — Agile 101 — <https://www.agilealliance.org/agile101/>