WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal Homepage: www.ijrpr.com ISSN 2582-7421

SIET Canteen Pre-Ordering and Queue Tracker Project Journal

S. Jeevanandham | N. Mariswaran | N. V. Pranith | Mrs. K. Sugashini

Student's, Department of Information Technology, Sri Shakthi Institute of Engineering and Technology Coimbatore, Tamil Nadu, India
* Project Guide| Assistant Professor Department of Information Technology, Sri Shakthi Institute of Engineering and Technology Coimbatore, Tamil Nadu, India

ABSTRACT

College canteens become crowded during short breaks, and students often spend a large part of the break waiting to order and collect food. Orders are taken manually at the counter, and students cannot tell whether their food is being prepared or is ready, so they stay near the counter and add to the congestion. This project presents QFlow, the SIET Canteen Pre-Ordering and Queue Tracker, a web-based application that lets students view the menu and prices, add items to a cart, place an order in advance and receive a token number for the order. The order status moves through the states Order Placed, Preparing and Ready for Pickup, and canteen staff update these states through a management interface. Menu data and orders are held in a Firebase cloud database that is shared by the student and canteen interfaces. The stored order records can also be used to review sales, popular items and busy ordering periods. The system is rule-based and does not use artificial intelligence or machine learning. The current implementation provides the home, menu, cart, order confirmation and login pages, and the screenshots in this journal show a menu of seven items and a working order and token flow. Functional and usability evaluation with students and canteen staff remains to be completed, and no measured reduction in waiting time is claimed.,

1. INTRODUCTION

College canteens provide food and refreshments to students, teaching staff and other employees. The time available is short. During breaks between classes and at lunchtime, many students try to buy food at the same time, which produces long queues and leaves little time to eat. Students may also reach their next class late because of the delay.

Web technologies make it possible to display a menu, accept orders before the customer reaches the counter and report the progress of each order. A single application that combines these functions may help reduce waiting and crowding, particularly where the break periods are short. QFlow applies this idea to the canteen of Sri Shakthi Institute of Engineering and Technology.

1.1 Existing challenges

In a traditional canteen, a student visits the counter, checks the available items, places the order and waits until the food is prepared. This is slow during peak hours, and manual order handling can lead to confusion, incorrect orders and delays when many customers are served at once.

A second problem is the lack of communication about order status. A student must stay near the counter to learn whether the food is still being prepared or is ready for collection, which increases crowding at the busiest time. Canteen management also has little organised information about which items sell most or when the rush occurs. Commercial food-ordering platforms are designed mainly for restaurants and home delivery, and they do not consider short break periods and quick collection at a college counter. A digital tool can only reduce delay if staff keep the menu and order status up to date, so its benefit depends on how consistently it is used.

1.2 Role of pre-ordering and queue tracking

Pre-ordering moves the ordering step away from the counter, so that preparation can begin before the student arrives. Queueing theory relates the number of customers in a system, their arrival rate and the time they spend waiting [1], [2], and it suggests that reducing the time each customer spends at the counter shortens the overall queue. A visible order status further reduces the need to wait at the counter, and a unique token for each order makes identification and pickup simple.

The approach used here is deliberately simple. It relies on a menu, a cart, an order record, a token and three order states, and it does not use prediction or learning. Such a system is easy to build, easy to explain to canteen staff and easy to maintain.

1.3 Problem statement and objectives

College canteens need a simple and reliable way for students to order before reaching the counter and to know when the order is ready, while giving canteen staff an organised way to receive and update orders. The objectives of QFlow are:

- To display the available food items and their prices in a web interface.
- To let a student select items, specify quantities, see the total amount and submit an order before reaching the counter.
- To generate a unique token number for every successfully placed order.
- To track and display the order status as Order Placed, Preparing or Ready for Pickup.
- To store menu details, orders, quantities and order status in a cloud database shared by the student and canteen interfaces.
- To provide a canteen management interface through which authorised staff manage the menu, view incoming orders and update their status.
- To keep order records from which sales, popular items and peak periods can be reviewed.
- To evaluate the application through functional, integration and usability testing.

2. MATERIALS AND METHODS

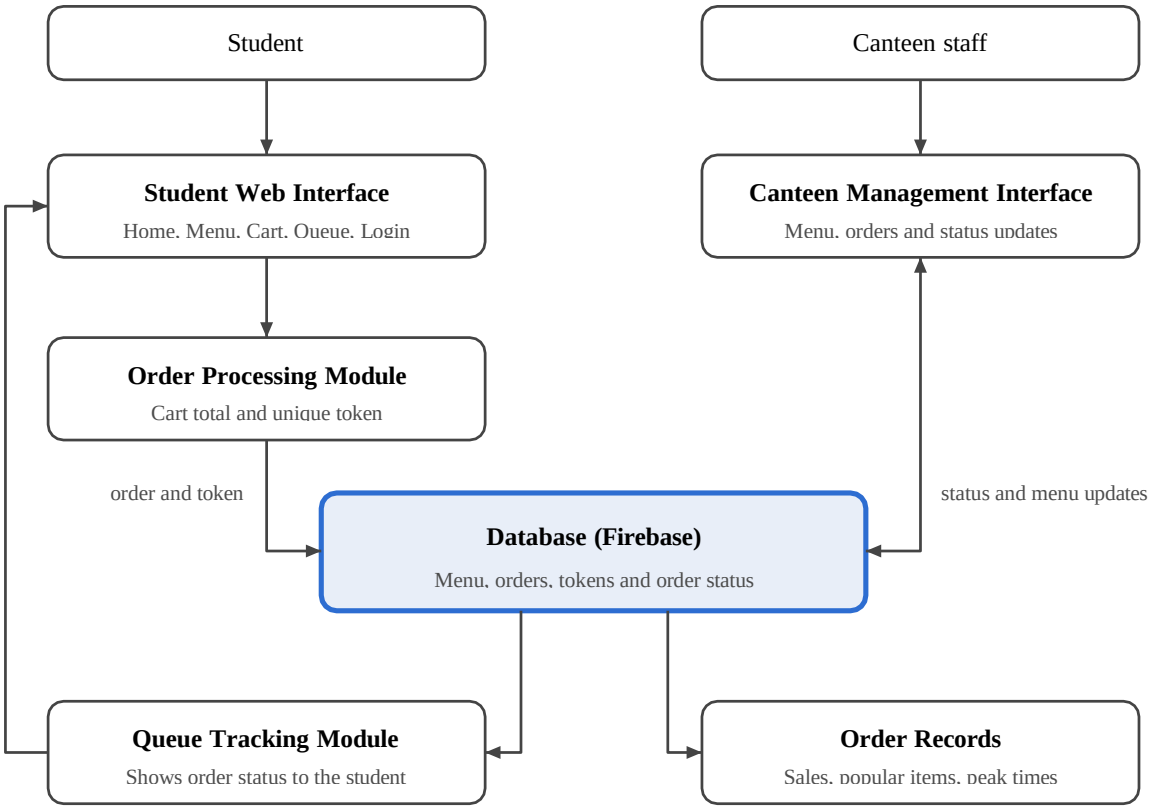
2.1 Development approach

The project followed a structured approach consisting of problem identification, requirement analysis, system design, implementation, integration and testing [3]. During problem identification, the difficulties faced by students and staff in the canteen during short breaks were considered. Requirement analysis identified the main functions: login, menu display, cart and order placement, token generation, queue tracking, and menu and order management by staff. The pages were then built and connected to the cloud database, and the order flow was checked in a web browser.

2.2 System architecture

QFlow follows a client and shared-database architecture. The student interface lets a student view the menu, build a cart and place an order. The order processing module calculates the total and generates the token, and the order is stored in the database. The queue tracking module reads the status of orders from the database and shows it to the student. The canteen management interface lets authorised staff maintain the menu and change the status of each order. Because both interfaces read and write the same database, a status change made by staff becomes visible to the student without manual communication.

Layer / component	**Responsibility**
Student Web Interface	Provides the Home, Menu, Cart, Queue and Login pages for viewing items, building a cart and placing an order.
Order Processing Module	Receives the submitted cart, calculates the total, creates the order record and generates the token number.
Database (Firebase)	Stores the menu, orders, tokens and order status, and is shared by both interfaces.
Queue Tracking Module	Reads order status from the database and displays it to the student.
Canteen Management Interface	Allows authorised staff to manage the menu, view incoming orders and update their status.
Order Records	Stored orders from which sales, popular items and peak periods can be reviewed.



2.4 Technology stack and development environment

Technology	Use in the project
HTML / CSS / JavaScript	Page structure, styling, cart handling and browser interactions.
Firebase	Cloud database for the menu and orders; the menu page shows “Loading menu from Firebase...” while the menu is retrieved.
Local development server	Used during development to serve and test the pages in a web browser (menu.html was served from a local address).
Web browser	Used to run and check the application.

The interface uses a navigation bar with Home, Menu, Queue and Login links on every page. [Authors: please confirm the exact Firebase service (Cloud Firestore or Realtime Database), the login method and the hosting platform, and edit this section to match the implementation.]

2.5 Menu data and cart calculation

The system processes only the information needed for its functions: login details, menu items and prices, the contents of each order, the order token and the order status. The menu in the current implementation contains seven items. The cart calculates each line amount as the item price multiplied by the quantity, and the order total as the sum of all line amounts. An empty cart shows a total of ₹0.

No.	Menu item	Price (₹)
1	Chappati	40
2	Chicken Rice	120
3	Noodles	90
4	Egg Rice	100

No.	Menu item	Price (₹)
5	Watermelon Juice	40
6	Lemon Juice	40
7	Tea	20

2.6 Database design

The database stores three kinds of records. A menu record holds the name, price and availability of a food item. An order record holds the items ordered, their quantities, the total amount, the identity of the user who placed it, the token and the current status. A user record holds the details needed for login. Menu records are read by the student interface and changed only by authorised canteen staff. Order records are created by the student interface, and their status field is changed by canteen staff. Only the information required for the application is retained, and the retention of old orders should be reviewed as the number of records grows.

2.7 Token generation and order status

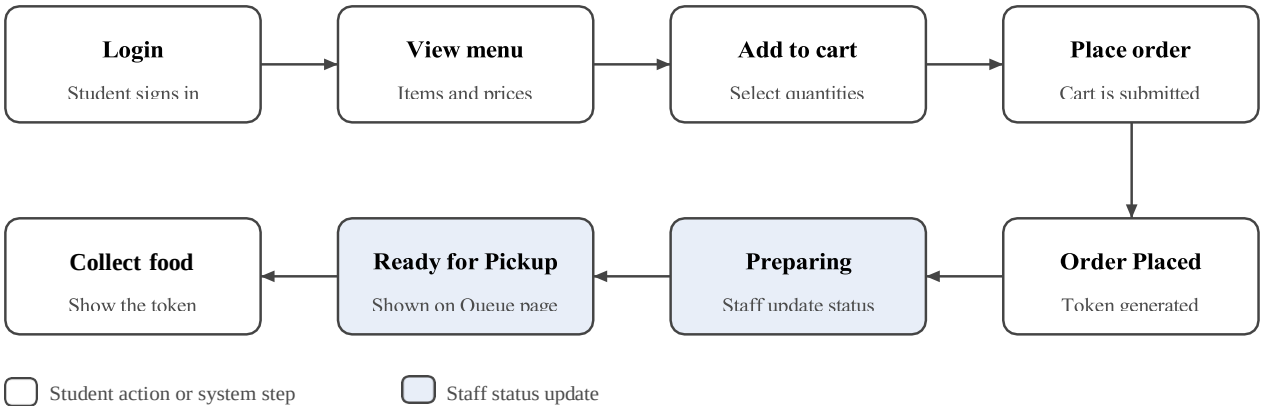
When an order is placed successfully, the order processing module generates a unique token number, which is stored with the order and shown to the student. The token identifies the order for the student and for the staff and is used to verify pickup. Each order then passes through three states, in the order shown in Table 2.

Order state	Changed by	Meaning
Order Placed	System, when the order is submitted	The order has been received and is waiting to be prepared.
Preparing	Canteen staff	The staff have started preparing the order.
Ready for Pickup	Canteen staff	The order is ready and the student can collect it using the token.

The current design relies on staff updating the status, so an order will show an outdated status if the update is delayed.

2.8 Order workflow

1. The student signs in and opens the Menu page.
2. The application loads the menu from the database and shows each item with its price and an Add to Cart button.
3. The student adds items; each addition updates the cart lines, quantities and the total.
4. The student can remove an item from the cart before ordering.
5. The student presses Place Order; the button shows “Placing Order...” while the order is stored.
6. A message confirms that the order was placed and gives the token number.
7. The order summary page shows the token, the ordered items, the total amount and the status Order Placed.
8. Canteen staff change the status to Preparing and then to Ready for Pickup, and the student collects the food using the token.



2.9 Testing and evaluation method

Unit testing can be used to examine functions such as the cart total calculation and token generation, while integration testing checks communication between the interface and the database. System testing evaluates the whole workflow, from login to collection. Usability testing examines readability, navigation and responsiveness on screens of different sizes [4]. The evaluation should also include error conditions such as invalid login, an empty cart and an unavailable network connection. This journal reports only behaviour visible in the screenshots of the implemented pages; no test pass rates, waiting times or response times are claimed.

3. RESULTS AND DISCUSSION

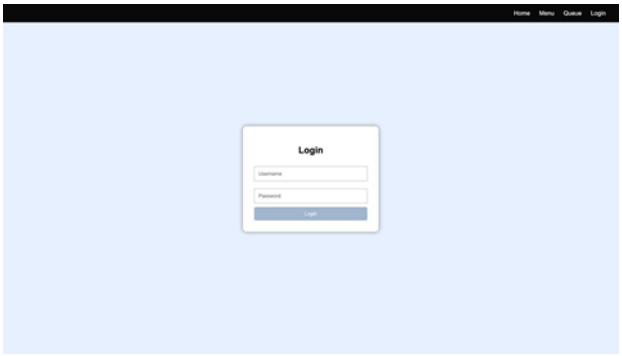
3.1 Implemented application features

The implemented pages bring the main steps of canteen ordering into one interface. The home page carries the message “Order Food. Skip the Queue.” and two entries, Menu and Q-Track (Fig. 3). The menu page lists the seven items with prices and an Add to Cart button for each, together with the cart and the Place Order button (Fig. 4). While the menu is being read from the database, the page shows a loading message (Fig. 5). The login page asks for a username and a password (Fig. 6).



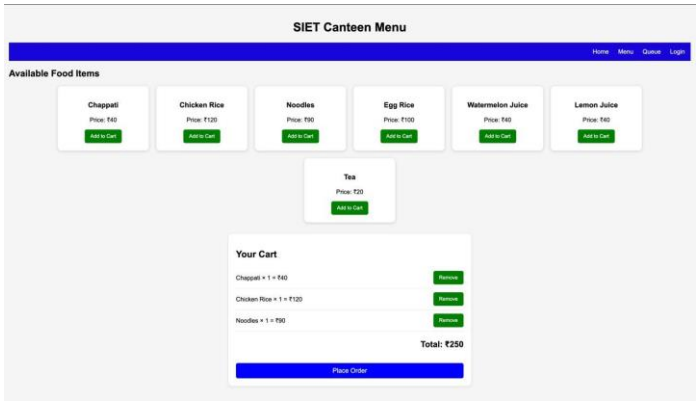


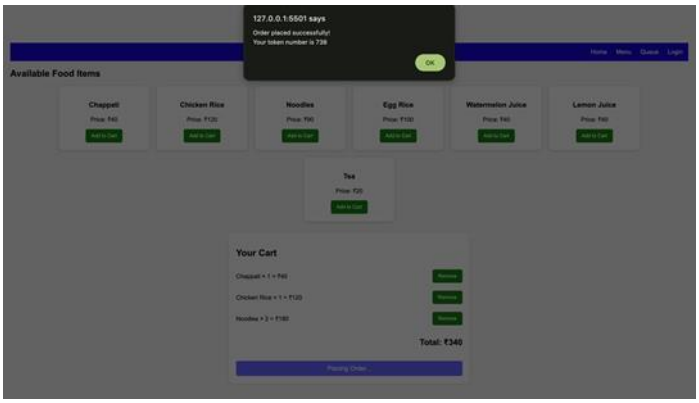




When items are added, the cart lists each item with its quantity and line amount and a Remove button, and shows the total (Fig. 7). In this example, one Chappati (₹40), one Chicken Rice (₹120) and one Noodles (₹90) give a total of

₹250. After Place Order is pressed, a message confirms that the order was placed and gives the token number (Fig. 8), and the order summary page shows the token, the ordered items, the total amount and the status Order Placed (Fig. 9). In the example shown in Figs. 8 and 9, the order of one Chappati, one Chicken Rice and two Noodles gives ₹340 and receives token 738.







The queue tracking page and the canteen management interface follow the order states in Table 2. Screenshots of the Preparing and Ready for Pickup states and of the staff interface were not available when this journal was prepared, so those parts are described as designed rather than demonstrated. The stored order records can be used to prepare summaries such as daily sales, popular items and busy periods; these summaries have not yet been evaluated.

3.2 Functional test scenarios

Table 3 lists representative test scenarios. “Observed” means the behaviour is visible in the screenshots of this journal; “Pending” means it has not yet been tested and should be reported after testing.

Test ID	Module	Test scenario	Observed / expected handling
T1	Menu	Open the Menu page	Observed: seven items with prices are displayed (Fig. 4); a loading message appears while the menu is retrieved (Fig. 5).
T2	Cart	Add items and check the total	Observed: lines, quantities and total are shown (Fig. 7); ₹40 + ₹120 + ₹90 = ₹250.
T3	Cart	Remove an item from the cart	Expected: the item is removed and the total recalculated. Pending.
T4	Order	Place an order with items in the cart	Observed: a confirmation message with the token number is shown (Fig. 8).
T5	Token	View the order after placing it	Observed: token, items, total and status Order Placed are shown (Fig. 9).
T6	Order	Place an order with an empty cart	Expected: the order is not placed and a message is shown. Pending.
T7	Token	Place two orders	Expected: each order receives a different token. Pending.
T8	Login	Sign in with valid and invalid credentials	Expected: valid user accepted, invalid user rejected with a message. Pending.
T9	Status	Staff change an order to Preparing, then Ready for Pickup	Expected: the student sees the updated status on the Queue page. Pending.
T10	Menu management	Staff add, change or remove a menu item	Expected: the student menu shows the change. Pending.

3.3 Comparison with conventional approaches

In the conventional approach, the student queues at the counter to see the menu, place the order and learn when it is ready. QFlow aims to separate these steps: the menu and ordering are available before reaching the counter, and the status is visible remotely. This may reduce the time spent at the counter and the crowding near it, and it gives staff an organised list of orders instead of a manual one. However, the system should not be considered faster or more effective than the manual process without a comparison of waiting times before and after its use and feedback from students and staff.

3.4 Discussion and limitations

Because the menu is stored in the database, items and prices can be changed without changing the pages, and the same approach can be adapted for school and office cafeterias. The system also has practical limitations. The accuracy of the order status depends on staff updating it promptly. The system does not include online payment, so payment at the counter is still needed unless it is added. The menu has no stock count, so an item may be ordered after it has run out unless availability is updated. A large number of orders in a short time requires load testing of the database and the interface. Because the application stores user accounts and orders, authentication, database access rules and protection of keys are necessary [5]. Further evaluation should measure waiting time, order accuracy and the satisfaction of students and staff.

4. CONCLUSION

QFlow presents a practical approach to reducing waiting and crowding in a college canteen. It combines a digital menu, cart-based pre-ordering, a unique token for each order, three-stage status tracking and a management interface for staff in a single web application, without using artificial intelligence. The menu, cart, order confirmation and token pages have been implemented and are shown in this journal. The effectiveness of the system depends on timely status updates by staff, the completeness of the stored records and further testing with real users.

Future development can focus on adding online payment, stock and availability control, notifications when an order is ready, a sales and peak-time dashboard for management, automated tests for the cart and token functions, and load testing during a full break period. A measured comparison of waiting times before and after deployment is required to determine the benefit of the system in practice.

5. ACKNOWLEDGMENTS

The authors thank Sri Shakthi Institute of Engineering and Technology for the academic environment and facilities that made this work possible. They are also grateful to the Head of the Department of Information Technology, to the project guide and to the faculty members for their advice and encouragement during the project.

6. REFERENCES

1. J. D. C. Little, “A proof for the queuing formula: L = λW,” Operations Research, vol. 9, no. 3, pp. 383–387, 1961.
2. D. Gross and C. M. Harris, Fundamentals of Queueing Theory, 3rd ed. New York, NY, USA: Wiley, 1998.
3. R. S. Pressman and B. R. Maxim, Software Engineering: A Practitioner’s Approach, 9th ed. New York, NY, USA: McGraw-Hill, 2019.
4. J. Nielsen, Usability Engineering. San Diego, CA, USA: Academic Press, 1993.
5. OWASP Foundation. OWASP Application Security Guidance. Available at: https://owasp.org/
6. Google Firebase. Firebase Documentation. Available at: https://firebase.google.com/docs
7. Mozilla. MDN Web Docs: HTML, CSS and JavaScript. Available at: https://developer.mozilla.org/
8. Sommerville, Software Engineering, 10th ed. Harlow, U.K.: Pearson, 2015.
9. G. J. Myers, C. Sandler, and T. Badgett, The Art of Software Testing, 3rd ed. Hoboken, NJ, USA: Wiley, 2011.
10. OWASP Foundation. OWASP Top Ten. Available at: https://owasp.org/www-project-top-ten/
11. Google Firebase. Firebase Realtime Database Documentation. Available at: https://firebase.google.com/docs/database
12. Google Firebase. Firebase Authentication Documentation. Available at: https://firebase.google.com/docs/auth
13. Mozilla. Web Storage API, MDN Web Docs. Available at: https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API