SSH – Joint Group Ordering Page

Engineering Design Review

Author: Mohammad Saad Mustafa
Student ID: 2673835
Date: 31/10/2024

Introduction

In a constantly evolving world where technology is deeply rooted into many different sectors, *Student Smart Homes (SSH)* and its wide range of products and services aim to revolutionize and modernize student living at accommodations all across the UK. SSH does this by offering an array of hardware and software solutions, such as the SSH App and SSH Console table, all of which work together simultaneously within a technological ecosystem.

Students living in shared accommodations often use grocery delivery services, placing orders collectively in order to split delivery fees and reduce costs. However, this process of collective ordering lacks a centralized system that manages the details of the orders and equally divides the total cost, making it challenging for students to view individual order details, determine which flat mate added what item, and calculate the cost of each respective purchase. Furthermore, tracking the fluctuations in prices of items becomes difficult.

Our proposal to combat this issue is to extend the SSH Console Table and the SSH App with the ability to add products from partnered supermarkets to a shared order for the next delivery. The aim of adding this extension to the existing ecosystem is to streamline and simplify the process of group orderings in a shared student accommodation. The new feature will be integrated into the SSH App as a page that will have an interface for students where they can view, add or remove products, edit their orders and personal details, and view the joint order with other students. The page will display information such as each students' individual items and respective costs. Unlike the SSH App, the console table will display information about the order from a group perspective, displaying details for all students participating in the current order as well as real-time tracking of a dispatched order.

Goals and Non-goals

- ***Goal:*** Reduce costs by at-least 35% for students when ordering in a group by splitting delivery fees. This will divide the initial cost of delivery among the number of participating students.
- ***Goal:*** Centralize the group ordering process for a more unified and streamlined experience.
- ***Non-Goal:*** Introduce subscription plans such as "Uber one" that take monthly payments and provide more services for subscribed users.
- ***Non-Goal:*** Promote any specific products from a supermarket to motivate students to buy them.
- ***Non-Goal:*** Allow for customization options in the application, such as theme selection and app icon changes.

Design overview

Our new group order page will be a sub-page on the SSH App. The page will be structured similarly to a food delivery application where there is a "main page" and all functions of the app can be accessed from there. The home page will run concurrently to all sub-pages where different processes of the group order will take place. We will liberally make use of WebSockets throughout our program to ensure real-time updates and communication between our clients and servers are efficient. The front end of the application will be in place to ensure that students using the service are provided with an efficient, interactive, and user-friendly interface. Whereas the back end will handle the data processing, data storage, interactions with the database, and communication with external factors like the partnered supermarkets and APIs. The topological layout of each page is:

Home Page – This will be the main section of the group ordering page and will act as a central hub from where all other pages can be accessed. At the top of the page, will be a navigation bar that will contain the search bar and menu to access the different subpages. All the UI/UX elements will be effective in providing students with a seamless experience. From this page, a student will be able to start a group order. Once they do, the other students in their accommodation will receive a notification and if they click on this notification, they will automatically be part of the order. Students can only join a group order within 4 hours of an order being created so that the delivery fee can be evenly and effectively divided at checkout.

Market page – A list view will be implemented here to display each product that is available to order, with each product showing details like its description and price. Users can browse products by selecting different categories, such as *"Fruits & Vegetables", "Dairy",* or *"Snacks"* using a collapsible menu that filters out products from the non-selected category and only display products from the selected category. Students can add a product by clicking "Add to cart" which will send the selected product to the connected "Review order/Checkout" page with the use of WebSockets to allow for instant updates to the cart .

The data for products will be fetched from partnered supermarkets using their API. The database will periodically fetch product lists and prices to keep data fresh and make sure prices are consistent with those on the supermarkets pages.

When adding a product to the group order, the API is invoked and the product to be added will have its "*Product_id"* from the "*Supermarket_details"* table added as an entry to "*Order_items"* in the "*Orders*" table. The opposite will be done when removing a product.

Review order/checkout page – On this page, Students can view a summary of the group order, including details about payment. The total amount that a student will pay will be calculated using the formula:

$$Cost = (\sum_{i=1}^{m} \text{Price i}) + \frac{\text{Total Delivery Cost}}{n}$$

Where,
- m is the number of items added by the student.
- Price i represents the price of each item.
- n is the total number of students participating in the current order.
- Total delivery cost is the shared delivery fees for the order.
- Constraints:
    - If m > 0 & Price i > 0, Cost > 0
    - n !> Total number of students living in the accommodation.

All data will be fetched from the database from "*Orders" and "Order_delivery" tables.* The page will comprise of a list view displaying all added items, Students can edit the order by removing or increasing/decreasing the quantity of products they added to the order. A "checkout" button will be available, and when pressed by all students participating in the order, will direct each student to the payment page where they can complete the transaction.

Personal page - This page will store personal information about the student such as, login details, address, contact information, and saved payment cards. All information on this page can be edited but will need verification through either the student's personal email or phone number. The sensitive information will be password protected and encrypted on our systems.

Current order tracking page - This page will be accessible after an order has been placed. If a partnered supermarket offers live tracking for the delivery, we will utilize a real-time map with GPS tracking to show the current location of the delivery vehicle as well as the estimated time of arrival. If a supermarket does not offer live tracking, then only the estimated time of arrival will be displayed.

Push Notification system – The app will send notifications to each student when another student has joined an ongoing order, create alerts when a group order has been created and display the time remaining to add or edit a student's individual cart, send reminders to each student to press the checkout button, inform students when the order is complete and they can pay, and alert students when their order is about to be delivered. We will utilize WebSocket's here to ensure that a student receives notifications instantly and does not have to refresh the app.

Console table display - For an ongoing order, a banner will be displayed that will show which students have pressed the checkout button and who hasn't. While an order is still ongoing, students who have not yet joined may do so by selecting their account through the "Add student to this order" option – this will only add them to that order, they will have to add products through the SSH App. The summary of the order and GPS map will also be available.

Existing data

Currently, we have access to the following data that already exists in the SSH system as well as data that we will need to create and implement when building our Group Order Page.

Table	Relevant Fields	Relevance
Student_account	Student_id, Student_name, Student_password, Student_address, Student_number, Student_mail, payment_details	Stores personal details for each student. - Student_id is a unique identifier for each student. Each student has one and this is the primary key for this field. - Student_password is sensitive information so is only accessible to the student and the database.
Orders	Order_id, Order_start_date, Order_price, Student_id, Order_items, Order_status	Stores details for all group orders. - Order_id is the unique identifier. - The Student_id is a foreign key here and stores the id's of all students participating in the order. - Order_items keeps track of all the items added to the group order and stores the Product_id, once the ID of a product is accessed, its subsequent details such as name, price and description may also be accessed.
Supermarket_details	Supermarket_id, Product_id, Product_name, Product_desc, Product_price, Supermarket_contsct	Records the details of each partnered supermarket and the products they list on the app. - Supermarket_id is the unique value given to each supermarket.
Order_delivery	Order_status, Order_id, live_tracking, Delivery_cost	Tracks the status of each order, including live tracking and estimated delivery time. - Order_status is a Boolean value where Order_Dstatus = "True" means the order has been delivered and Order_Dstatus = "False" means the order has not been delivered.

Existing role and access control

SSH already has built-in user roles for both students and administrators, each with defined permission and access control. Students have access to the Personal page, Home page, Marketplace page, Review order page, and Track order page. They do not have access to the data stored in the database. Administrators like accommodation admin staff can oversee all orders within their accommodation. They can monitor the orders and intervene if an issue arises.

Alternatives

Alternative use of the Console table

Allowing students to place orders through the Console table. Instead of only being able to join an order from the console table, this feature would've given students access to adding products or editing their carts from the console table itself.

Pro - This feature would've increased accessibility, allowing students to easily place orders from a central interface. This would aid us in streamlining the group ordering process.
Con – Allowing students to edit their orders through the console table will increase the complexity of the group ordering function. Since a new interface for the Console table would be required, it will require additional testing, development, and maintenance.
Con – This feature would've raised security and privacy concerns. Due to the position of the console table, sensitive information like payment information or items that a student might want to order discretely might be less secure on the shared console table.

Implementing this feature would have increased the complexity of our program as well as increased the privacy and safety concerns.

Alternative way to start a group order

If a student wants to start a group order, they can send a unique link generated by each new group order to other students in the accommodation. Once the link is pressed, the student pressing it will be added to the order.

Pro – This method to start a group order would be easy to use as students can just click the link and join it, instead of searching up the order or asking someone to add them.
Pro – Sharing a link would ease cross-platform flexibility. Since a link is generated, students may join using any device they wish as most, if not all, devices can access a URL link.
Con – Security concerns arise with this method, as a link could be shared with someone who is not part of the accommodation or administration, allowing unauthorized users to access the order. Since all SSH technologies are interconnected through the SSH Cloud, if the SSH app is compromised and a hacker gains access, they could potentially obtain unauthorized access to students' personal data.
Con – For the student starting the group order, it may be a hassle or tedious task to send the link to each student that wants to participate. Some accommodations have more than 10 students residing in them so remembering to send it to all students may not be a feasible option.

This way to start an order is not feasible as it would be reliant on students sending links for group orders. Therefore, being prone to human error. Instead the method we implemented is automated through the app and notification system thus being less error prone.

Milestones

***Milestone 1*: Develop & Implement backend logic for joint group order and Notification system.**

- Develop the backend logic that will enable the students to use the feature and give our program functionality. With the backend operating, students will be able to place group orders. Once the system to start a group order is complete and functional, we will develop the notification system, alerting students in the same accommodation when a group order has been created so that they can interact with it and join the group order. The team will use WebSockets to instantly broadcast real time updates to all students in the group order.

- To test this feature we will create a joint group order, then we will test the notification system and ensure that the notification is sent to the right students, within the expected time limit. Pressing the notification should allow a student to join the order. At this stage, students will not be able to add or remove products to the order.

***Milestone 1b:* Develop pricing system**

- With the use of WebSockets, we will develop a system that dynamically calculates the cost each student must pay at checkout. WebSockets will allow students to see, in real time, the price updating as a product is added or removed from the cart.

***Milestone 2:* Integrate the Marketplace interface and API.**

- Integrating the backend with the marketplace interface will allow it to communicate with the partnered supermarkets' API, allowing students to search and access products listed on the supermarkets API directly from the SSH app.

- We can test this by adding and removing products from a shared order to ensure all actions are running as expected

***Milestone 3:* Implement and integrate order tracking system and logic.**

- Develop the logic to handle the GPS tracking and estimated delivery times for orders from partnered supermarkets. The frontend element of a live map will be integrated with the back end to provide a fully functional order tracking page feature.

- To test this feature, we can coordinate with the partnered supermarkets that provide live tracking and arrange dummy deliveries to ensure that the GPS tracking is accurate.

***Milestone 4:* Develop the Frontend user interface.**

- Since we have implemented the backend, we now need to implement the UI so that students can view and manage joint orders. The UI should enable students to add/remove products, edit the quantity of products, view descriptions of each product, and view the cost of the total order as well as what each individual student will pay.

- To test this feature, we will set up a beta test with a closed group of students. Since we are designing this feature for students, their feedback would be most impactful and relevant to improving the UI and functionality of the feature. If significant issues are reported by the closed test group, we can back track and fix those issues. We will repeat this process of closed group beta testing until the issues are resolved.

Milestone 5: **Integrate order summary and tracking display on Console table.**

- Develop the Console table interface to display a summary of the most recent or currently ongoing order, live order status, estimated delivery time, and participant details. The students should also be able to join the current ongoing order from the table.

- We will test this feature to ensure that the order details are accurately reflected on the console table screen and are responsive when students engage with it.

Milestone 6**: Final testing and Deployment.**

- We will conduct extensive testing on real data to ensure that the system can handle a large number of students using it simultaneously without performance degradation. Furthermore, we need to test the same for students participating in a group order.

- Once the testing has been completed and is successful, we can integrate the shared order feature into the main SSH App and console table, making it available to all students.

Dependencies

- ***UI/UX Design team:*** Responsible for designing the user interface for the group ordering feature on the SSH app, including all sub-pages and functions, such as the Marketplace page, Personal page, Current order page, and the Console table interface.

- ***Backend Development Team:*** Responsible for developing, implementing, and integrating the logic for the backend of the application, as well as management of the database to support the group orders. They will need to implement the logic for live order tracking and integrating it with the partnered supermarkets' APIs. Furthermore, since we will require new data fields in our existing database schema, the backend team will be responsible for implementing these in the existing database and making them available for use.

- ***Notification Systems Team:*** Required to create and manage notification system for order invitations, reminders, and status updates to be sent automatically to students in a shared accommodation.

- ***Testing & Quality Assurance Team:*** This team will be in charge of running tests on the different functions of the app and ensure that the quality of the code and overall program is of the best quality.

- ***Security Team:*** Since sensitive data, such as credit card information, will be used the security team will need to ensure that strong encryption methods are being deployed so that all sensitive data is protected and securely stored.

Cost

Significant increases in costs are not expected due to the new joint group ordering feature as the feature will leverage SSH's existing infrastructure to keep incremental costs low. The tasks we will be performing, such as data processing or queries made related to group orders would be minimal and can be managed on the existing SSH servers. Since security will be a priority, ensuring sensitive data is protected may require periodic investments into additional encryption protocols as transactional volume and users grow. Furthermore, beta testing and quality assurance checks will be required, but the overall costs are expected to remain manageable, with minimal long term expenses.

Privacy & Security concerns

The new feature will require the collection of new sensitive data, such as payment methods. Therefore, privacy and security methods will be pivotal in maintaining data security and integrity. Such sensitive data will be securely stored

using hashing algorithms, such as bcrypt, which will ensure that even if a data breach occurs in the database, the passwords remain encrypted. During data and payment transmissions, we will use Secure socket layer (SSL) or Transport layer security (TLS) protocols to encrypt the connection between students and the servers. We will also need to obtain permission from students to store the new sensitive data in our database.

The key security concerns that are a threat to our design are:
- A data leak or breach in the database: since we will be processing and storing students' sensitive data, we will need to ensure that access to data is maintained and regulated.
- We will also need to ensure that unauthorized users are not able to access and join a group order. This will compromise and risk the users in that order.

Risks

Risks	Mitigations
Students might not consent to sharing their data	We can provide a privacy policy that, in great detail, explains how their data will be collected and used in our systems.
Unauthorized users might access student accounts	To prevent this, we must use strong password conventions and make sure data is thoroughly encrypted with our hashing algorithms.
Orders might be incomplete upon delivery or not reach on time	Establish customer relations with partnered supermarkets so that students may contact them incase an issue with their order arises.
High traffic on the server could lead to performance issues	Use caching strategies to manage the traffic to/from servers, ensuring that increased demands is controlled

Supporting material

- Auth0. (n.d.). Hashing in action: Understanding bcrypt. from https://auth0.com/blog/hashing-in-action-understanding-bcrypt/
- DigiCert. (n.d.). What is SSL, TLS, and HTTPS? Retrieved October 31, 2024, from https://shorturl.at/4j5LQ
- PubNub, (n.d.). WebSockets: A complete guide on WebSocket connections. Available at: https://www.pubnub.com/guides/websockets/