Student Smart Homes - Shared order feature

Engineering Design Review

Author: Jakub Nita
Date: 23 October 2024

Introduction

Grocery shopping can be difficult for the average student. Many students who own cars cannot bring them to university due to inadequate parking space in student areas, and those that do bring their cars may struggle to find the time of day to go on a grocery shop with so many deadlines to meet. Grocery delivery services have largely solved this issue for students - but they aren't perfect. None of these services provide a feature where students living together can also place orders together. If this were to be implemented, ordering groceries would become much more convenient for students; less time would be spent waiting around for deliveries as there would be less of them, less money would be spent on delivery and service charges, and students could qualify for multibuy offers on items they commonly need. From an environmental standpoint, this would also reduce the carbon footprint associated with multiple trips from delivery services.

This is why we want to extend the SSH Console Table and SSH App with a shared order feature, accessible with a new section of the SSH App labelled 'Shared Orders'. This feature would allow students to view information from our partner supermarkets about available products along with their current prices and add them to a shared order for the next delivery (prices will change over time as promotions come and go or the base price of a product changes). Students would also be able to view the items in the order at any time, including who added what, what the total cost is, and the total cost of all items for each individual member of the house. Users will be able to view their delivery radius showcasing all of the supermarkets they may order from, as well as the opening and closing times for delivery of each supermarket in the radius. In order to actually place the order, housemates contributing to the order will have to show their approval for it with an 'Approve' checkbox in the SSH App, and one of them will have to press a new 'Place Shared Order' button at the SSH Console Table touch screen once all students with items in the order have shown their approval. The SSH Console Table will then use its existing feature to automatically split the bill between all contributing students using their saved payment methods.

Adding this feature aligns with our goals to make student homes easier to live in and to free up time for students to focus on their academic work, and entering the grocery delivery market will almost certainly increase our profits due to the low cost associated with undertaking this project. User engagement will also increase providing more opportunities for in-app purchases and advertisement viewership, thus furthering our revenues even more.

Goals and non-goals

- **Goal:** Implement ‘Shared Orders’ section of SSH App where students can share grocery orders with their flatmates. 15% of all active SSH App users should have placed an order using this feature in the first 6 months of release.
- **Goal:** Integrate ‘Shared Orders’ in SSH App with SSH Console Table to add ‘Place Shared Order’ button on SSH Console Table touch screen.
- **Goal:** Establish partnerships with 5 of the following supermarket chains before release: Tesco, Sainsbury’s, Asda, Morrisons, Aldi, Co-op, Lidl, Waitrose, Marks & Spencer.
- **Non-goal:** Partnering with takeaways and restaurants.
- **Non-goal:** Making a separate app to provide this service.

Design overview

Upon selecting the ‘Shared Orders’ feature in the SSH App, students will be greeted by a JavaFX GUI containing their delivery radius on a map. The radius will contain all of the supermarkets they can order from and their locations on the map. Upon selecting a supermarket, some text will pop up; the text will contain either a green ‘Open’ or red ‘Closed’ header to signify the supermarket being opened or closed for delivery, alongside the opening and closing times for delivery for each day of the week underneath the header. Open supermarkets will also contain an ‘Order’ button which, once pressed, views a comprehensive list of available products for that supermarket and their prices. Multibuy offers and discounts will be displayed next to each relevant product, and users will be able to filter the products shown based on the type of product (e.g. fruit and vegetables, bakery, household), and other factors such as ‘ready to eat’ and ‘offers’. Furthermore, students will be able to search for specific items using a search bar. Once a user presses on a product, further information will be listed such as a product description, nutritional values, ingredients, and allergen information. Finally, next to each item will be an ‘Add to order’ button which adds that item to the shared order. All of the information on product availability, pricing and everything else will be retrieved from partner supermarket databases using the Java Database Connectivity (JDBC) API.

Let’s say Students 1, 2 and 3 all live in House A and Student 1 wants some items from Sainsbury’s in Selly Oak. Once Student 1 has added the items they want, the SSH App will automatically send out a push notification to Student 2 and Student 3, letting them know that Student 1 is ready to make an order. This will be handled using the Apple Push Notification service (APN) for iOS users and Firebase Cloud Messaging (FCM) for Android users. If Students 2 and 3 do not want anything, then they simply let Student 1 know and Student 1 can decide if he wants to place the order himself or wait for the others to need groceries. If Student 1 urgently needs the items, then he will be able to place the order on his own. Now let’s say Students 2 and 3 want to add some items to the order. Upon opening the GUI to view the delivery radius, a ‘Current Orders’ button will be available to press somewhere on the screen. Once this button is pressed, a list of all current orders for House A will pop up - in this case Sainsbury’s Selly Oak will pop up (this feature should be able to display multiple supermarkets as housemates may want to order from different places). Next to the name of Sainsbury’s Selly

Oak there will be the same ‘Order’ button previously mentioned, where students can view products and their prices and add them to the order, alongside a ‘View Order’ button which will show a list of each item in the order as well as who added what, what the total cost is (including the delivery fee), and the total cost of all items for each individual member of the house.

Once Students 1, 2 and 3 have all added their items and are happy with the order, they must all tick a checkbox labelled ‘Approve’ in ‘View Order’. Students in the house who did not add any items to the order do not have to do anything. Once they have ticked ‘Approve’, one of them must go to the SSH Console Table and press the ‘Place Shared Order’ button on the touch screen. The Console Table will then use its ability to make purchases from third-party providers and automatically split the bill to do exactly that, using the saved payment methods for each student contributing to the order. Deliveries will be handled in a similar fashion to Just Eat, where anyone can become a driver (provided they fulfill the necessary requirements such as having a valid driver’s license), and where drivers are paid for each order they complete. Each student should only pay for what they added, and the delivery cost should be split based on the percentage of the order total their items contributed to. The charge for a particular student can be calculated using this simple method in Java:

```
public static double calculateStudentCharge(double studentTotal,
double orderTotal, double deliveryCost) {
   double percentageContribution = studentTotal / orderTotal;

   double deliveryShare = percentageContribution * deliveryCost;

   return studentTotal + deliveryShare; }
```

Note that orderTotal does not account for the delivery charge i.e. it is only the cost for all of the items.

If any of the students do not have a saved payment method, the tablet will prompt those students to make a one-time payment for that order or to add a payment method to their account and use it.

Alternatives

The SSH App does not currently have a feature which allows it to make purchases from third-party providers and automatically split the bill unlike the SSH Console Table, hence why the order has to be placed from the SSH Console Table tablet. Implementing this feature for the SSH App would allow students to place orders remotely which would be useful particularly when students are returning back to university and need to prepare a grocery order for the first day back. However, this would raise security concerns as payment information would have to be stored in the SSH App which can be stolen using malware on student phones or through the phone being physically stolen, whereas the SSH Console Table is more robust to hacking and is more difficult to steal as it does not move from the student homes.

Moreover, the step where all students in the order must tick the ‘Approve’ checkbox makes the ‘Shared Orders’ feature more time consuming and less convenient as students may forget to tick the box and housemates would have to chase them up about this. Alternatively, we could omit the checkbox. The downside here is that bad communication between housemates may lead to one member of the house placing the order before the others are ready (before the others have added all of the items they want, etc).

Milestones

- **Milestone 1:** Write all of the code necessary for the new feature in preparation for internal testing.

- **Milestone 2:** Conduct internal testing using TestFlight for our iOS app and the Google Play Console for our Android app. Check that the ‘Shared Orders’ GUI in the SSH App and ‘Place Shared Order’ button in the SSH Console Table are working. Also check that the JDBC API used to retrieve information from supermarket databases is working properly using a test database.

- **Milestone 3:** Establish partnerships with supermarkets as outlined in ‘Goals and non-goals’. Only move on if we manage to establish these partnerships.

- **Milestone 4:** Roll out the new versions of the SSH App and SSH Console Table tablet to users.

Dependencies

- **UI team:** Will need to design the GUI that appears when accessing the new ‘Shared Orders’ feature of the SSH App. The button for the SSH Console Table tablet will have to be added as well.
- **Java team:** Will need to write all of the code linking the GUI with the JDBC component which the database team will handle.
- **Database team:** Will need to make sure partner supermarket databases are connected successfully to the JDBC component and that they can be queried.
- **Notifications team:** Will need to ensure Apple Push Notification service and Firebase Cloud Messaging are configured to send push notifications out to housemates when a student is ready to make an order.
- **Human resources:** Will need to implement the system for becoming a delivery driver, tracking driver deliveries, and for paying drivers.

Cost

We do not expect operating costs to increase significantly as a result of Shared Orders. While we will have to increase spending on cloud resources to run servers to enable the feature to

work, this is fairly inexpensive and we will make use of a metered cloud service to increase spending only when needed.

Privacy and security concerns

Users will not have to consent to any new data being used as Shared Orders only uses existing data. Nevertheless, we must be careful when designing the new feature to ensure we are not creating new security vulnerabilities. The JDBC component for example should be robust to SQL injection.

Risks

Supermarkets might have varied APIs or integration requirements	Code can be modified to include these APIs and requirements
Securing partnerships with supermarkets may take longer than expected	We could contact the supermarkets as soon as the EDR is accepted (if it is)
Use of JDBC introduces the risk of bad actors using SQL injection to steal data etc.	We can implement input validation and parameterised queries to prevent SQL injection
‘Approve’ checkbox might drive users away from using the feature	This can be removed.
Users may dislike having to use Console Table touch screen to use ‘Place Shared Order’ button	We can integrate this button into the SSH App.

Supporting material

Food Standards Agency - Food and You 2: Wave 4 Key findings Chapter 7 (2022)
https://www.food.gov.uk/research/chapter-7-food-shopping-sustainability-and-environmental-impact