Task 2: Design Justification

**Designing Effective Output:**

‘Keep display simple’ and ‘keep presentation consistent’ are two concepts of ‘Designing Effective Input’ that are used in the design. This is achieved through consistent use of legible fonts with typographical hierarchy, colours, paddings, margins, navigation and button style throughout. The design facilitates user movement through the use of clear indications of the next step to take, e.g. buttons at bottom of the page, and a sticky progress indicator at bottom of screen. What makes the design visually attractive is the use of Google’s Material design principles, such as elevation, shadow, and simple colour (and use of Google Sans font). In terms of colour and its psychology, the design is primarily green, often associated with freshness and good taste (red has been associated with hunger, but it is often associated with fast food. The app aims to be associated more with freshness than fast food).

**Designing effective input:**

Forms easy to use:

Menu item quantity has large, simple + and – buttons to indicate quantity, with the current quantity displayed in between. This has been chosen for ease of use over a text-input or drop-down selector. The quantity selectors also have an instruction bar. There is a comment box before order is submitted to allow students to add requests or comments to their order. This has a caption above a multi-line input field so that it is obvious to the user what the box is for.

Screen flows:

Flows left to right and top to bottom, because this is how we are used to reading and consuming information. Information is in the same logically-grouped areas on each screen and the screens flow fairly seamlessly between each other. Each screen only deals with one step of the ordering process. All of these considerations aim to reduce cognitive load for the user. An example of logical grouping is on the restaurant list page, where each restaurant has its own white box, clearly delineating each list item.

**Human-Computer Interfaces:**

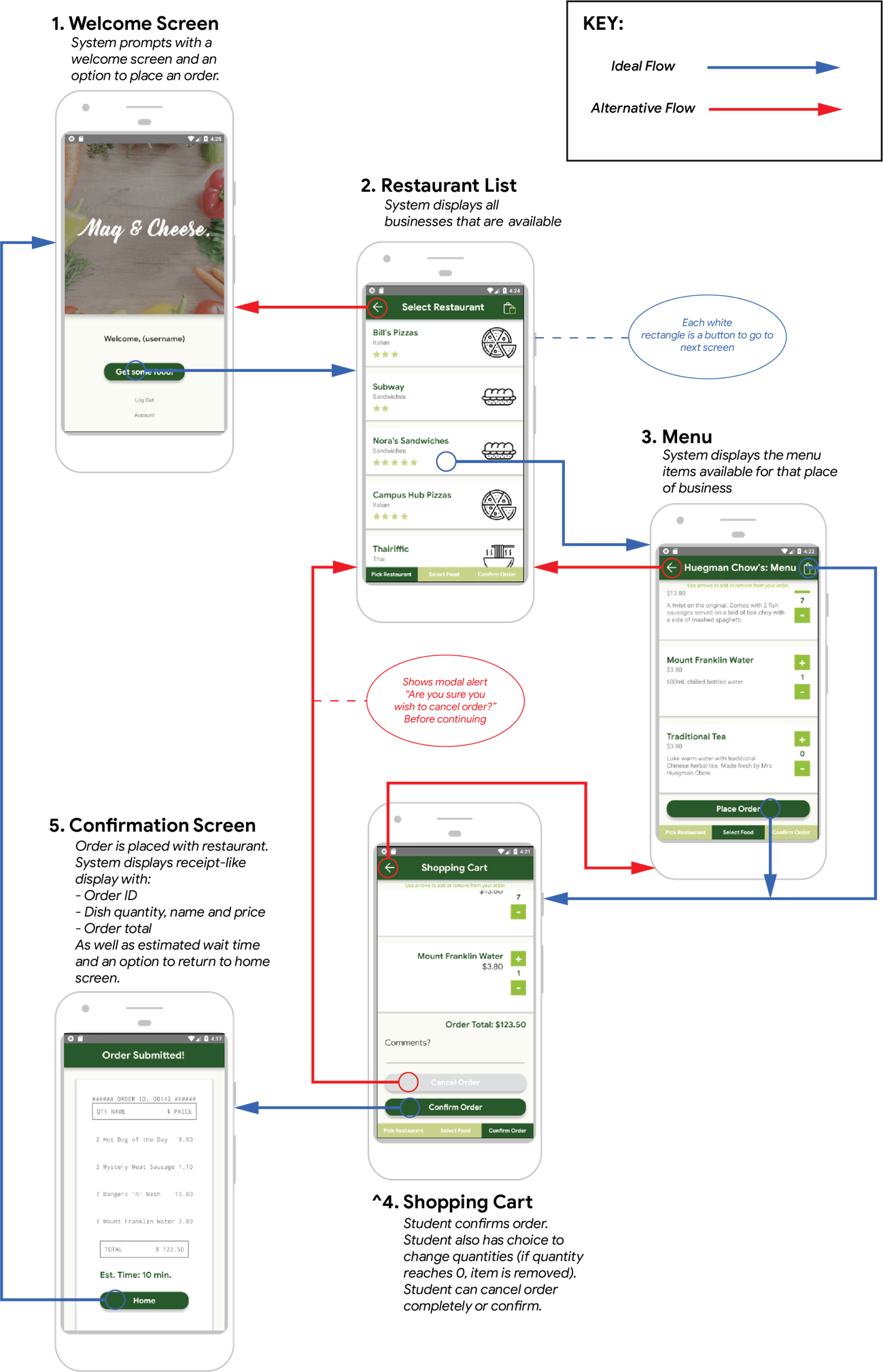
UI is efficient:

The design uses simple colours, design and graphics to ensure load speed is optimal. There is comfortable spacing of elements without wasting screen space. The ordering process is broken up into many screens to reduce loading time, i.e. versus one monolithic screen. The FlatList React Native element is used frequently in the design. This element can work similarly to AJAX - it loads more elements as they are required, aiding in efficiency.

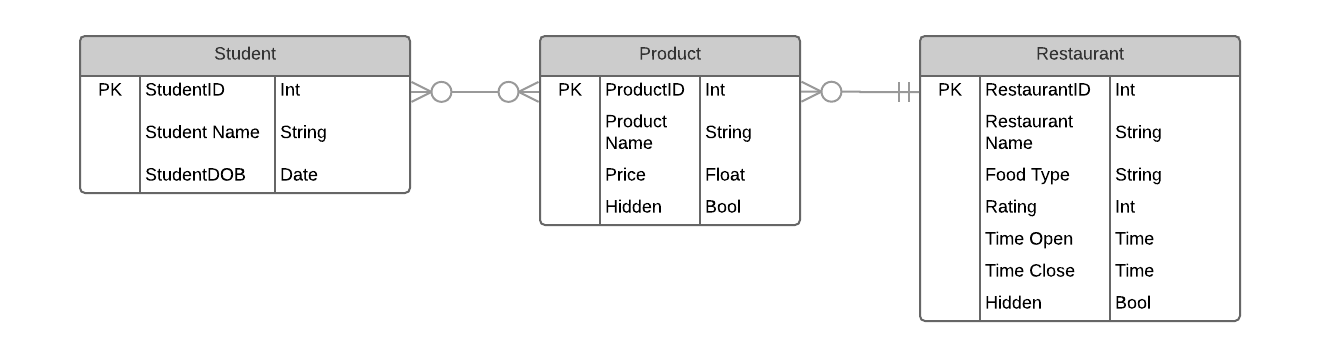
Buttons are simple in design and there is an obvious difference between the primary action and the secondary action, i.e. “Submit Order” is green, while “Cancel Order” is grey. If ‘cancel order’ is selected, a modal will appear to confirm this action.

\*\*\* I know this is longer than half a page, but it is broken up for legibility which makes it take up more space

Task 3: Storyboard:



Task 4: ER Diagram



**Justifications:**  
Student -  
The student entity uses a StudentID primary key for verification and identification. We assume the software will ultimately connect to the Macquarie system to pull this information at account creation. The student’s date of birth is included for instances where alcoholic beverages can be ordered via the application.

Product -

Products are unique to restaurants and identified through the ProductID. While there maybe duplicates for items like Mount Franklin water, discrepancies in pricing make it difficult to standardize items for all restaurants. Due to this, restaurant menus are generated from the product table and their inclusion is toggled with the Hidden attribute.

Students can purchase multiple products and a product can be bought by multiple students. The many to many, open relationship between Student and Product makes an associative table (Orders) necessary. It is visible in the second diagram.

Restaurant -  
Restaurants are identified by the Restaurant ID. Products can be assigned to only one restaurant and it must exist before a product can be created.  A rating evolves over time from student feedback represented in the application as stars (out of 5) and as an int between 0-10 in the database (0 and half stars included). The time open and close attributes are included for display in the application and to prevent students making orders to closed restaurants. A hidden attribute has also been including in cases where a restaurant would like to suspend itself from taking orders with the intention of returning in the future.

Task 5: Database Tables

