Bei dem Design des Frontends wurde sich für eine einfache und klare Ausrichtung entschieden. Die verwendeten Farben beschränken sich auf eine kleine Palette von weißen, grauen und schwarzen Farben. Lediglich für das Logo, welches auffallen soll, wurde in Orange-Töne ausgesucht.

Auch das Layout aller Sites wurde einheitlich programmiert. Zu beginn jeder Seite findet sich im „Header“ das Logo der Food-Truck-App als ImageButton wieder. Es dient dem Benutzer gleichzeitig als bequeme Möglichkeit in das jeweilige Hauptmenü zurückzukehren – angemeldete Kunden werden in die „CustomerMenuActivity“ und Betreiber in die „OwnerMenuActivity“ navigiert. Dies wird erreicht, indem jede Activity im Programm eine Funktion „getToHome“ besitzt, welche die aktuelle Activity zur Hauptmenü-Activity wechseln lässt.

Im Operator-Zweig der Anwendung befindet sich im Header zusätzlich ein Zurück-Button, welcher die Benutzererfahrung weiterhin verbessert. Der Header wird vom eigentlichen Inhalt jeder Site durch einen Divider, einer horizontalen Linie in grauer Farbe, getrennt.

Die nachstehende Abbildung 3.2.1 zeigt das beschriebene Grundlayout einer jeden Site der Food-Truck-App.

Es handelt sich also insgesamt um ein funktionales Design, welches dem Benutzer bei seiner Navigation durch das Programm unterstützen soll und die Anwendung einfach zu verstehen macht.

Wie bereits in der Architekturbeschreibung des Frontends unter „2.2.3 Frontend“ beschrieben, spaltet sich die Anwendung in die Zweige Customer und Operator. Die Grundfunktionen für Login und Registrierung teilen sich die Zweige jedoch, da diese nicht speziell auf eine Benutzergruppe zugeschnitten sein müssen. Um beim Login/Registrierung zwischen einem Kunden und einem Betreiber unterscheiden zu können, wird der Nutzer in einer vorgestellten Ansicht aufgefordert anzugeben, welcher Benutzergruppe er angehört.

Bei einem erfolgreichen Login wird der Kunde bzw. Betreiber in sein jeweiliges Hauptmenü navigiert (Siehe Abbildung „3.2.2 Hauptmenü Customer“ und „3.2.3 Hauptmenü Operator“).

**CustomerMenuActivity**

Ausgehend von der CustomerMenuActivity (Hauptmenü eines Kunden) erreicht man Activities für das Aufgeben einer Bestellung, das Nachverfolgen der eigenen Bestellungen und das Anzeigen der Truck-Infos des Betreibers sowie eine Activity, mit welcher man sich einen Standort des Food-Trucks vorschlagen lassen kann, indem man seinen eigenen angibt.

**// TODO: Customer…**

**OwnerMenuActivity**

Ausgehend von der OwnerMenuActivity (Hauptmenü eines Betreibers) erreicht man Activities für die Routenplanung, das Anpassen der Speisekarte, das Einsehen der Bestellungen, die Lebensmittelbestellung, als auch für Funktionen die den aktuellen Standort des Betreibers betreffen.

**OwnerRoutebearbeitenActivity**

= Haupt-Activity zur Routenplanung. Hier wird die aktuelle Route in einer ListView angezeigt, welche die Namen sowie die Ankunfts- und Abfahrtszeiten der Standorte enthält. Mittels eines Buttons „Standort hinzufügen“ erreicht man eine weitere Activity „OwnerAddstandortActivity“. Durch Klicken eines Elements der Liste erreicht man eine Activity „OwnerStandortbearbeitenActivity“, gleichzeitig übergibt man das Element der Liste.

**OwnerAddstandortActivity**

= Unter-Activity zum Hinzufügen eines Standorts. Hier finden sich Eingabefelder für den Namen, den Breitengrad(X), den Längengrad(Y), die Ankunftszeit und die Zeit in Minuten, welche man an dem Standort verbringen möchte. Mittels eines Buttons „Standort hinzufügen“ wird eine Funktion ausgelöst, die den Standort anlegt.

**OwnerStandortbearbeitenActivity**

= Unter-Activity zum Bearbeiten oder Löschen eines ausgewählten Standorts. Hier wird das übergebene Element in die Eingabefelder analog der „OwnerAddstandortActivity“ gefüllt und kann bearbeitet werden. Mittels eines Buttons „Standort speichern“ wird eine Funktion ausgelöst, die die Änderungen speichert. Mittels eines weiteren Buttons „Standort entfernen“ wird eine Funktion ausgelöst, die den Standort löscht.

**OwnerSpeisekarteActivity**

= Haupt-Activity zur Bearbeitung der Speisekarte. Hier wird die aktuelle Speisekarte in einer ListView angezeigt, welche die Namen und Preise der Speisen enthält. Mittels eines Buttons „Speise hinzufügen“ erreicht man eine weitere Activity „OwnerSpeiseNeuActivity“. Durch Klicken eines Elements der Liste erreicht man eine Activity „OwnerSpeiseBearbeitenActivity“, gleichzeitig übergibt man das Element der Liste.

**OwnerSpeiseNeuActivity**

= Unter-Activity zum Hinzufügen einer Speise. Hier finden sich Eingabefelder für den Namen und den Preis einer Speise. Mittels eines Buttons „Zutat hinzufügen“ erreicht man eine separate Activity zum Erfassen einer Zutat und der geforderten Menge. In einer ListView werden die angelegten Zutaten angezeigt. Mittels eines weiteren Buttons „Speise anlegen“ wird eine Funktion ausgelöst, die die Speise anlegt.

**OwnerSpeiseBearbeitenActivity**

= Unter-Activity zum Bearbeiten einer ausgewählten Speise. Hier wird das übergebene Element in die Eingabefelder analog der „OwnerSpeiseNeuActivity“ gefüllt und können bearbeitet werden. Dies gilt auch für die ListView der zum Gericht angelegten Zutaten. Mittels eines Buttons „Bearbeitung abschliessen“ wird eine Funktion ausgelöst, die die Änderungen speichert.

**OwnerBestellungenActivity**

= Haupt-Activity zur Ansicht der Bestellungen. Hier werden von Kunden aufgegebene Bestellungen in einer ListView angezeigt, welche die IDs und Status der Bestellungen enthält. Durch Klicken eines Elements der Liste erreicht man eine Activity „OwnerBestellungActivity“, gleichzeitig übergibt man das Element der Liste.

**OwnerBestellungActivity**

= Unter-Activity zur Detailansicht einer Bestellung. Hier wird das übergebene Element in die TextViews für Uhrzeit, ID, Standort, Person und Status gefüllt. Diese können nicht bearbeitet werden. Ebenfalls wird eine ListView geladen, welche die bestellten Speisen inkl. Preis anzeigt. Mittels eines Buttons „Zubereitung beginnen“ wird eine Funktion ausgelost, die den Status der Bestellung auf „in arbeit“ setzt. Mittels eines weiteren Buttons „Zubereitung abschliessen“ wird eine Funktion ausgelost, die den Status der Bestellung auf „erledigt“ setzt.

**OwnerLebensmittelbestellungActivity**

= Haupt-Activity zur Lebensmittelbestellung. Hier wird die aktuelle Anzahl der Bestellungen pro Gericht in einer ListView angezeigt, welche die Namen und Bestellzahlen der Gerichte enthält. Zusätzlich zu der Anzahl der Bestellungen lässt sich ein Zuschlag pro Gericht eingeben. Mittels eines Buttons „Lebensmittel bestellen“ wird eine Funktion ausgelöst, die ermittelt, welche Zutaten pro Gericht gefordert sind und multipliziert diese mit der geforderten Menge an Gerichten.

Mittels eines weiteren Buttons „zeige alle Zutaten“ wird diese Anzahl der Zutaten in einer separaten Activity angezeigt, ohne die Lebensmittelbestellung auszulösen.

**OwnerAktuellerstandortActivity**

= Haupt-Activity für Funktionen, die den aktuellen Standort betreffen. Hier wird mittels Eingabefeldern und Buttons das Auslösen der „Reservierungsannahme“ und des „Reservierungsschluss“ in „X“ Minuten ermöglicht.