



ASHESI UNIVERSITY COLLEGE

**CREATING A PEER-TO-PEER FOOD VENDING, ORDERING AND
DELIVERY ECOSYSTEM IN COLLEGES.**

APPLIED PROJECT

B.Sc. Computer Science

Ronald Akai Nettey

2018

ASHESI UNIVERSITY COLLEGE

**CREATING A PEER-TO-PEER FOOD VENDING, ORDERING AND
DELIVERY ECOSYSTEM IN COLLEGES.**

Applied Project

Applied project submitted to the Department of Computer Science, Ashesi
University College in partial fulfilment of the requirements for the award of
Bachelor of Science degree in Computer Science

Ronald Akai Nettey

April 2018

DECLARATION

I hereby declare that this applied project is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

.....

Candidate's Name:

.....

Date:

.....

I hereby declare that preparation and presentation of this applied project were supervised in accordance with the guidelines on supervision of applied projects laid down by Ashesi University College.

Supervisor's Signature:

.....

Supervisor's Name:

.....

Date:

Content

DECLARATION	3
Content	iv
List of Tables	vi
List of Figures	vi
Acknowledgement	vii
Abstract	viii
Chapter 1: Introduction	1
1.1 Background	1
1.2 Motivation	2
1.3 Proposed Solution	2
1.4 Related Works and Literature.....	3
Chapter 2: Requirement Specification	7
2.1 Project Overview and Scope	7
2.2 Requirement Gathering	8
2.3 User Characteristics	8
2.4 Functional Requirements	9
2.5 Non-Functional Requirements.....	16
2.5.1 Performance Requirements	16
2.5.2 Security Requirements	16
2.5.3 Compatibility Requirements.....	16
2.6 Design and Implementation Constraints	17
Chapter 3: Architecture and Design	18
3.1 High Level Architecture	18
3.1.1 Presentation Layer	19
3.1.2 Business Logic Layer	19
3.1.3 Data Access Layer	19
Chapter 4: Implementation	20
4.1 Application Development Process Model	20

4.2	User Interface Design	20
4.4	Tools and Technology	21
4.4.1	Adobe XD.....	21
4.4.2	Adobe Illustrator	22
4.4.3	Ionic Framework	22
4.4.4	Google Firebase Products.....	23
4.5	User Interfaces	24
4.5.1	Authentication pages	24
4.5.2	Home Page for the Customer Application	25
4.5.3	Customer App – Food Store.....	26
4.5.4	Vendor Rating Page.....	28
4.5.5	Cart Page	29
4.5.6	Checkout Page	30
4.5.7	Delivery Request Page	31
4.5.8	Delivery Request Acceptance Page	32
Chapter 5:	Testing and Result	33
5.1	Unit Testing	33
5.1.1	Block 1 – Fetching of Data from The Cloud Storage and Populating the Products Page. 33	
5.1.2	Block 2 – Gradle Update	34
5.1.3	Block 3 – Ionic Native Plugins Inclusion.....	34
5.2	System Testing.....	34
5.3	User Testing and Feedback.....	35
5.3.1	Feedback.....	35
Chapter 6:	Conclusion and Recommendation	37
6.1	Summary	37
6.2	Limitations and Challenges.....	37
6.3	Future Work.....	38
6.4	Conclusion	39
References.....		41

List of Tables

Table	Description
Table 2.1	Requirements Gathering
Table 2.2	Use Case: View Menu from Vendor
Table 2.3	Use Case for Placing an Order
Table 2.4	Use Case for Making a Delivery Request
Table 2.5	Use Case for Subscribing to a Vendor

List of Figures

Figure 3.1	High Level Architecture
Figure 4.1	Authentication Pages
Figure 4.2	Home Page For Customer Application
Figure 4.3	User Interfaces for The Food Store of a Selected Vendor.
Figure 4.4	Rating Interface for A Selected Vendor
Figure 4.5	Cart UI
Figure 4.6	Checkout Page User Interface
Figure 4.7	Delivery Initiation Pages
Figure 4.8	Delivery User Confirmation User Interface

Acknowledgement

I will like to thank the Lord God Almighty for seeing me through all this. It has not been easy at all but by the motivation and encouragement given to me by parents and friends I have been able to make it this far. I would like to express my gratitude to my supervisor Mr. David Ebo Adjepon-Yamoah, for his patience, dedication and efforts he placed in making this project a success. I would also like to use this opportunity to thank everyone who supported and believed in the success of this project. Much appreciation goes to Manuela Doh, Elorm Dartey, Edel Togobo, Tamisha Segbefia, Joseph Mills, Derick Omari and Dzifa Hodey for their immense support and contribution in making this project a successful one. Thank you.

Abstract

Motivated by the current state of technological advancement, this project seeks to modernize the traditional methods of food vending, ordering and delivery by creating a unique yet satisfying mobile experience for users. It is with no doubt that some tools and technology exist that may be employed. However, in the case of student run food business, the default platform used to receive orders is via an embedded google form in an email broadcasted to the entire Ashesi University College community. In several cases, most student food businesses set up at a location on campus and advertise as much as possible to gain a larger reach of customers and traffic to their store/location. Although these methods solve the problem of getting students to patronize their products, there are several efficient ways such methods could be improved to gain a larger reach of customers. This project seeks to develop a peer to peer mobile application that grants users the choice and convenience of ordering, vending and delivering food items using their smart phones.

Chapter 1: Introduction

1.1 Background

Food plays a very important role in our lives, because of its importance, its availability when needed is crucial. Food joints such restaurants, cafeteria, among others, currently provide readily available sources of food close our homes, workplaces and places of interest. Placing order for your meal from your preferred restaurant only to wait for hours for the delivery staff to arrive late during peak periods. Another reason for the late arrival is the consistent occurrence of food orders appearing at the same time which also need to be delivered. In addition, delivery staff of the various restaurants may be dormant when there are less orders placed in a day (Liu, et al., 2017). All these challenges exist in the food delivery economy. Thus, triggered the interest to take advantage of the widespread of mobile devices, the internet and advancement in technology to provide a solution to these challenges.

The concept of the Food mobile application came about, as a solution to the inconvenience of not being able to identify what meals are present at the student cafeteria. Swarmed with a ton of work from students all the way to the staff and faculty of the Ashesi University College, time has become as essentially vital. This subtly poses a lot of threats on our health as it deteriorates our eating patterns as students, staff and faculty.

1.2 Motivation

With the massive penetration of mobile technology and internet usage in Ghana, these devices can be used to improve the experience of ordering, vending and delivering food. We are often faced with the challenge of long waiting hours, high delivery fees and geographic barriers (in terms of how far our preferred restaurants are). The delivery paradigm is drastically shifting and developing a wider reach of audience using technology and the internet.

As students of Ashesi University College, we are consumed by lucrative and laborious tasks geared towards developing our standard of living and impacting our community positively. Thus, time is a crucial and expensive resource we extensively try to manage therefore we cannot afford to disrupt our ability to withhold as much of it. It is with no doubt that the rate of growth of the student population keeps increasing. Between 2014 and 2018, 171 students out of 880 were enrolled in the year 2014 (Ashesi University College, 2018) and 275 freshmen in the year 2018 (Ashesi University College, 2018) forming an increase rate of 60% over the 4-year period. The increase in population coupled with the number of cafeterias available will only increase the discomfort that already exists with meal ordering and vending especially at peak periods.

1.3 Proposed Solution

The solution proposed by conducting this project is to develop a food ecosystem. A series of applications that work together to improve the experience of users who sells food items and users who purchase food items as well. Two mobile applications will be developed, the vendor version of the mobile application and the customer version of the application. The third feature is a delivery module that would be embedded in both applications. The main idea of this project is to make use of the increasing population of the Ashesi community in association with the

Opportunistic Online Takeout Ordering & Delivery service (O-OTOD). This specifically makes use of the people around to incidentally deliver food to their peers upon request (Liu, et al., 2017). The vendor version of the application should allow users manage order, products and subscribed customers within the mobile application. The customer version should allow users access stores of vendors around them and place orders from these stores. The delivery module should allow the customers place delivery requests to their peers around the vendor and have them delivered to the customer by a fellow member of the community who accepted the request. Inadvertently, the project also promotes the entrepreneurial drive of users as it creates a lucrative environment for a user to only fulfil delivery requests at a fee to accumulate some revenue. The termed used for the name of the mobile application in this project is coined as FoodApp.

1.4 Related Works and Literature

Online on-demand food delivery services are rapidly emerging in our current century with many vendors shifting from the local paradigm of serving customers to the use of technology to improve customer experience and patronage. According to Technavio's market study in the "Global Online On-Demand Food Delivery Services Market 2017-2021", convenience and ease of accessibility of food items are one of the essential growth factors for online on-demand food delivery services (Technavio, 2017). Considering that, with the increase in adoption of the internet and improvement in technology, several vendors and companies have taken this as an opportunity to tap into the ever-growing food delivery market to increase customer patronage.

According to the government of Ghana, in a report published by Jumia ^[1], Ghana's mobile subscription rate continues to burgeon with 34.57 million mobile subscriptions in total, forming a total percentage of 129% between 2015 and 2016. Also, Ghana's mobile penetration rate lies at a percentage of 119% with one third of the country's population (10.11 million) being active internet users ^[1]. In addition, these service and applications are usually used by millennials which forms majority of the primary end users of these food delivery services. Identifying the tedious lifestyle of these millennials and other consumers as such, several vendors have stuck to the approach of employing order-focused food delivery services and logistics-focused food delivery services which are made accessible via the use of mobile and computer devices. According to the market analysis done by Technavio, the food delivery market will grow steadily at a compound annual growth rate of close to 32% by 2021 (Technavio, 2017).

Moreover, the global online on-demand food delivery service has a few most important stakeholders in the market, making it highly competitive. "It has the presence of a few major players and many small- and mid-size vendors." (Wire, 2017). Considering how competitive, the market of online on-demand food delivery systems or services are, the novelty of this paradigm is evident and its efficiency directly proportional to the adaptation rate of mobile devices in such geographic regions.

" Technology penetration—mainly smartphone and online penetration—has only been slightly relevant to the speed of adoption so far due to the geographic expansion of food players. We believe that the food category will grow in line with the smartphone category as new smartphone users adapt their behavior to take full advantage of the technology." (Hirschberg, Rajko, Schumacher, & Wrulich, 2016).

The availability of technology and its advancement has created a lucrative environment for companies to reach a larger audience and for clients to get the best of experience. Moreover, the food delivery industry is being improved rapidly, offering customers the power to order from home or wherever they are out of a variety of restaurants. According to the Business Wire, a press release distribution and regulatory disclosure company, four major vendors played a major role in the contribution of about 10% to 12% of the global revenue in 2016 (Wire, 2017). Such vendors included: Delivery Hero, Foodpanda, GrubHub, and Just Eat. These companies have not just adapted to the use of technology but also reshaped their business model to suit the needs of the users due to the same expectancy they desire after being accustomed to shopping online with the use of smartphone applications or websites.

Several applications and platforms have been developed with the aim and objective of disrupting the way users obtain food. Correspondingly, Delivery Hero is among one of the major companies that has made an investment in the food ordering, vending and delivery space by providing an online platform that connects customers to the nearby restaurants. In an interview conducted by the McKingsey Quarterly, Niklas Östberg, the CEO of the company mentioned that the platform managed to acquire the loyalty of its users by aiding in the identification of available and respectable restaurants. Additional functionalities on the platform include the analytics on savings, past orders review and online payment.

Delivery Hero employs both the classic delivery model (aggregation) and the new delivery model as discussed to serve its users. The strength of this platform as compared to this project is its use of big data to tailor specific restaurants to specific or general users. An example stated by Niklas (2016) is the use of data to link customer service staff to restaurants when there more feedback from customers is negative (Schumacher & Swinford, 2016). The goal of the

platform is to disrupt ineptitudes in delivery and the inefficiency in manual processes as stated by Niklas (2016). As it stands, the success of the platform is burgeoning and was valued at 42 million Euros in 2013 to 297 million Euros in 2016 forming an outstanding 607% increase between 2013 and 2016 (Statistica, 2017).

To curb this problem, there have been several solutions generated by individuals in this community. Some of which include, a student community website which features a section for the current menu at the student cafeteria which is updated by students. In addition to that, there has been several attempts to provide intelligent solutions to address this discomfort. Some of which include a Telegram channel which updates users on the current menu at the student's cafeterias. Students could also place orders by sending their messages to the channel. Also a smart solution was proposed by (Sylvia Engman, 2017) which involved the use of a mobile application which allowed users to purchase food online. However, as it lies, these discomforting situations of food ordering and vending still lurk. As outlined in the previous paragraph, although these projects serve as great solutions to this problem, there exists certain limitations that prevent the solutions from making a lasting impact.

Chapter 2: Requirement Specification

This chapter elaborates into detail the various aspects and functionality of the application, the FoodApp goes on to highlight the features of the application, the underlying technologies that make it operational, the various interfaces of the system, the functions of the various components of the system and the operational conditions of the application. In addition, the purpose of this chapter is to provide a clear documentation on the various functions and components of the application to stakeholders or developers.

2.1 Project Overview and Scope

The FoodApp is an application developed in two folds. The first type of application represents the vendor's side of the platform whereas the second type of the application merges the customer side of the application and the delivery system. The focus of this paper will be mainly on the customer side of the application with reference given to the vendor's application where necessary. Giving an overview of the features of the FoodApp mobile application, it allows users to view available vendors near them, shop from their preferred vendor, add products to cart from multiple vendors, checkout items in cart with their preferred method of payment (payment methods should include: mobile money, the Ashesi meal plan and payment on pick up) and the ability to place delivery requests. Built with the Ionic Framework, the Food® application has one codebase which is developed and compiled into an android and iOS application with ease. The combination of all these features are directed towards the user's need to curb the wastage of time, explore the menu of other food vendors near them and to gain some active income during their leisure time on campus.

2.2 Requirement Gathering

TABLE 2.1 – REQUIREMENTS GATHERING

Resources	Information to be gathered	Methodology
Students that run food ventures on campus.	User specifications	Information will be obtained exclusively by interacting with these students.
Students that use the cafeterias and food joints on and off campus.	User needs and specification	Information will be obtained exclusively by interacting a random set of students.
Similar applications	<p>The features users were well pleased with.</p> <p>The features that displeased the users.</p> <p>The drawbacks of the applications.</p> <p>The reason behind the non-continuity of these applications.</p>	<p>Retrieving the feedback obtained by the developers of these applications (if they were created by students).</p> <p>Testing similar applications.</p> <p>Interacting with users on these applications in the Ashesi University community.</p>

2.3 User Characteristics

The user must have a smart phone able of establishing a connection to the internet and as well capable of granting access to the current GPS coordinates of the device. The application

is designed with not more than four colors which makes it suitable for users with aesthetic affinity. In addition, an article written by Syed Ghayas et al (2013), the recognition rate of how users perceive graphical elements, vary by how familiar these representations are to the users (Syed Ghayas, 2013). Considering that, the application is designed with a modernistic outlook and familiar icons and graphics for the youth in general and developed for all age groups. The application may not be suitable for users concerned about their location. The application functions majorly by using the location of the devices to load the appropriate contents. By using the application, one agrees to this term and condition.

2.4 Functional Requirements

Table 2.2 - Use Case: View Menu from Vendor

Use Case: View Menu from Vendor	Primary Actor: Customer(Foodie)
Description:	This use case describes the generic function of the application which allows users to view the menu of preferred vendors currently present on the application.
Goal:	To display of items from the selected vendor
Success Measurement:	The user being able to view the corresponding menu items present within the vendor's store.
Precondition:	If the user is a new user of the application, authentication is not needed. The user has the

	liberty of viewing any menu from a vendor without any restrictions but cannot purchase just yet.
Trigger:	The user taps the vendor present in the vendor's list.
Event Flow:	<p>Once the mobile application is opened:</p> <ol style="list-style-type: none"> 1. The user is welcomed with a guest account. 2. The list of vendors is displayed on the homepage. 3. The vendor selects a preferred vendor 4. The list of items present on the vendor's menu is displayed.

Table 1.3 – Use Case for Placing an Order

Use Case: Place an order	Primary Actor: Customer(Foodie)
Description:	This use case describes the customer's(Foodie's) process of ordering a meal from a one or multiple vendor.
Goal:	Place an order from one or more vendors.
Success Measurement:	A confirmation screen displays after confirmation of a successful order.

Precondition:	For the vendors to identify their clients, the user will have to sign up if he/she does not have an account or sign in if otherwise. This is to also secure transactions and maintain integrity. In addition, payment must be made as well as a delivery option selected before proceeding to place the order.
Trigger:	User taps the checkout button
Event Flow:	<p>In the cart page:</p> <ol style="list-style-type: none"> 1. The business logic checks to see if the user is a guest. If yes, the user is prompted to sign in or sign up. Upon completion of authentication, the cart is reloaded with the data stored in the local storage of the mobile device. 2. The user selects a delivery method 3. The user selects a payment method. 4. The order is processed, and all necessary checks are made to ensure the validity of the order. Thus includes: communicating with the database to confirm the availability of the item at that time and the

	<p>validity of the payment pin typed in by the user.</p> <p>5. Should any of the conditions be false, the order placement is halted otherwise, successful and displays the success page.</p>

Table 2.4 – Use Case for Making A Delivery Request

Use Case: Make a delivery request	Primary Actor: Customer(Foodie)
Description:	This use case describes the delivery request functionality as used by the user to place a delivery request to delivery Foodies after placing an order from a vendor.
Goal:	Place a delivery request and get a response from a Foodie nearby the vendor from which the items were purchased from.
Success Measurement:	The accepted delivery Foodie is displayed in the checkout page of the user before completing the checkout process.

Precondition:	<p>The user must be logged in and must grant the application permission to access its location.</p> <p>Items must be present in the cart of the user.</p> <p>There should be enough funds in the user's account to proceed.</p>
Trigger:	<p>The user selects the “Make Delivery Request” option in the dropdown menu on the cart page.</p>
Event Flow:	<p>Once the “Make Delivery Request” option has been selected:</p> <ol style="list-style-type: none"> 1. The application asks for the user's permission to access and get the current location. (This is important to allow the delivery users deliver the items to the user accurately.) 2. Once the location has been obtained, the request is sent out to other Foodies near the vendor and waits for a response.

	<ol style="list-style-type: none">3. If there is no response after 1min, the request times out and the user has the option of making the request again.4. If there is a response, the profile of the delivery user who triggered the response is displayed with the estimated time of arrival and delivery cost inclusive.5. The user (Foodie) has the option of declining the current delivery user and placing a new request if not satisfied or accepting the delivery user.6. The user is redirected to the updated cart page which now includes the delivery user as well as the delivery charge added to the total.7. The user (Foodie) then pays and checks out.
--	---

Table 2.5 – Use Case for Subscribing to a Vendor

Use Case: Subscribe to a vendor	Primary Actor: Customer(Foodie)
Description:	This use case describes the social feature of the application which allows users to subscribe to vendors. This allows them (Foodies) to receive notifications about new products, when vendors come online, and promotions set by the vendors.
Goal:	Subscribe to vendors
Success Measurement:	The “Subscribe” button on the vendor changes to “Unsubscribe” and the vendor gets added to the subscribed list of vendors in the user’s account.
Precondition:	The user must be signed in.
Trigger:	The user taps the “Subscribe” button on the vendor’s page.
Event Flow:	<ol style="list-style-type: none"> 1. Select a preferred vendor. 2. Display the profile of the vendor. 3. Tap on the “Subscribe” button.

2.5 Non-Functional Requirements

2.5.1 Performance Requirements

The android and iOS mobile applications are built using one codebase with the use of the Ionic Framework. This framework using web technologies to create hybrid applications and has numerous plugins that enables the application to use native features without any disturbance or crashes. The application runs seamlessly on both devices. However, some animations lag and distorts the user experience when using CSS animations. To curb that, angular (framework on which Ionic is built on) is used to improve the flow of animations on the mobile devices. To minimize long waiting periods when fetching data from the database, infinite scroll is used. Infinite scroll allows the user to load items that can fit on the screen of the device. Upon scrolling, it fetches another set of items rather than waiting till the dataset has been loaded into the application.

2.5.2 Security Requirements

With the use of Google's Firebase Authentication module, the credentials and data of the user are kept secured. The user will have to be authenticated before performing any form of transaction on the application.

2.5.3 Compatibility Requirements

The Ionic framework allows for one codebase, multiple builds on different platforms. The application is compatible with android devices with Android 6 (API level 23), iOS devices as well as Microsoft devices. However, the application will only be built for android devices and iOS devices.

2.6 Design and Implementation Constraints

- Android version should be or above Android version 6 (API level 23) as most of the plugins used in the application require it to function properly.
- Enough memory to store needed information.

Chapter 3: Architecture and Design

This chapter focus on the system architecture of the Food@ application. In addition, the technologies used, structural components, user interactions and design will be expounded on.

3.1 High Level Architecture

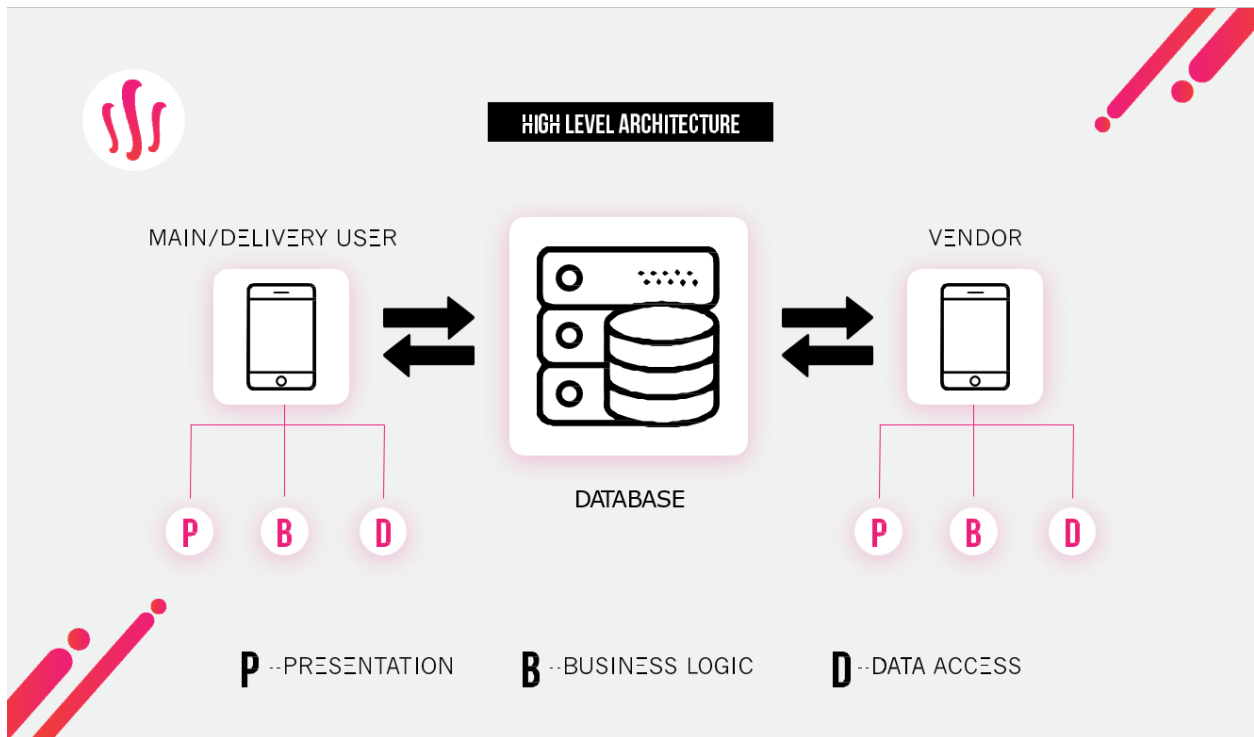


Figure 3.1 – High Level Architecture

The FoodApp will be developed based on the three-tier architecture. This architecture is a client-server architecture pattern of which the user interface of the application is referred to as the “Presentation”, the logical processes and calculations as the “Business Logic” and the storage and access of data as “Data Access”. In this architecture, the presentation, business logic and data access are developed and maintained independently as modules on the specific devices or platforms.

3.1.1 Presentation Layer

This represents the various interfaces the user interacts with on the application to perform tasks.

3.1.2 Business Logic Layer

Calculations and processes of data are performed in this layer with data passed from the presentation layer as input. Data is processed and parsed based on the constructed logic in the application to yield meaningful information that is either displayed to the user or communicated to the database layer.

3.1.3 Data Access Layer

The database server to be used mainly for this application is Google Firebase. It allows for data to be persisted and accessed in real-time.

Chapter 4: Implementation

Variety of technologies have been used in the development of this mobile application. This chapter focuses on the application development methods and processes, tools and software used in the implementation of the FoodApp. In addition, the major interfaces of the application will also be discussed in this chapter.

4.1 Application Development Process Model

One of the main goals of this application is to create a comfortable experience for the user while solving their problems. As such, the scrum agile model was used to meet the requirements of the users by collaboratively working on various aspects of the application and establishing an iterative and incremental development. Modules of the application was developed and tested, based on the feedback given, the project was iteratively worked on thus leading to the incremental development of the application with result being a high-end quality mobile application which satisfied the user's needs.

4.2 User Interface Design

Much focus and time was dedicated to the design of the user interface. Attention to details was key and was not taken lightly as the focus was to grant the user the best of experience when using the application. The colors, layout and animations were developed and improved upon based on the feedback received from earlier versions of the application. A minimal approach was taken in designing the application with not more than four colors and font types used. In addition to the development and design of the user interface, the “Eight Golden Rules of

Interface Design” developed by Shneiderman (2016) was adhered to. In light of that, in order to produce an interactive and properly designed interface, following the eight golden rules, the interfaces were designed with consistency: same color scheme and font types on all pages, universal usability, informative feedback, dialogues that yield closure, error prevention designs, easy reversal of actions, control to the user and a reduced short term memory load (Shneiderman, 2016).

4.4 Tools and Technology

This section elaborates the tools and technologies used in the development and implementation of the FoodApp.

4.4.1 Adobe XD

Developed by Adobe systems, Adobe XD is used to develop a great user experience using a variety of tools and techniques. It supports vector design, wireframe design and has a prototyping feature which allows users to test designed user experiences on devices. It will be used to design the interfaces and user experience. With the prototyping feature, users can use the generated prototypes and give feedback which will be used in the iterative design. (Adobe, Introducing Adobe XD., 2018)

4.4.2 Adobe Illustrator

This software was also developed by Adobe Systems with the purpose of creating vector designs. In this project, this tool will be used to generate vector-based designs such as logos and icons. (Adobe, The state of the art of illustration., 2018)

4.4.3 Ionic Framework

The ionic framework is an open source Html5 mobile application development framework which is used to develop hybrid mobile applications. It is built on top of Angular JS and Apache Cordova. It's hybrid nature allows for the development of one codebase which can be run on any platform. It will be used to build the android application with the help of HTML5, SASS and Typescript. (Ionic, 2018)

4.4.3.1 HTML 5

HTML is the acronym for Hyper Text Markup Language which is used to develop the structure of a webpage. The current version of HTML5 is used in this project. It is used in this project to develop the structure of the user interfaces in the Ionic application. (Mozilla, 2018)

4.4.3.2 SASS

SASS is the acronym for Syntactically Awesome Stylesheets, a preprocessor scripting language which is used to easily generate Cascading Style Sheets when compiled. It is used in this project to style the developed structure of the application (SASS, 2018).

4.4.3.3 Typescript

Typescript is a superset of JavaScript mainly used in Ionic and includes optional static typing to the language. It was developed and is maintained by Microsoft. It is used mainly in this project in the business logic layer of the application for computations. (TypeScript, 2018)

4.4.3.4 Angular 4

Angular is the framework on which Ionic is built on. It is an open source front-end framework developed by Google as a solution to the challenges involved in developing single page applications (Angular, 2018).

4.4.4 Google Firebase Products

Google firebase has a range of products that are used in this project to develop a highly interactive application. (Google, 2018)

4.4.4.1 Cloud Firestore

A no SQL database hosted on the cloud which allows the storage and syncing of data between users and devices. It will used to store user data on the FoodApp.

4.4.4.2 Cloud Storage

A loud based storage platform that allows various form of user-generated content to be stored. In addition, file transfer and download are secured with Google Security. It will be used to store mainly user generated images in the application.

4.4.4.3 Authentication

Firebase authentication has multiple authentication methods which can used to securely manage users. Aside the casual email and password authentication it also provides third party authentications with providers such as Google or Facebook. It will be used in this project to securely manage user authentication in the application.

4.4.4.4 Cloud Messaging

This product allows for notifications and messages to be sent out to users on various platforms. This tool will be used to handle every form of notification to be sent out by the Food® Application.

4.4.4.5 Android 6 SDK

Provides the necessary tools to build an android application kit. It is used by Ionic to generate an APK after a build to an android platform.

4.5 User Interfaces

4.5.1 Authentication pages

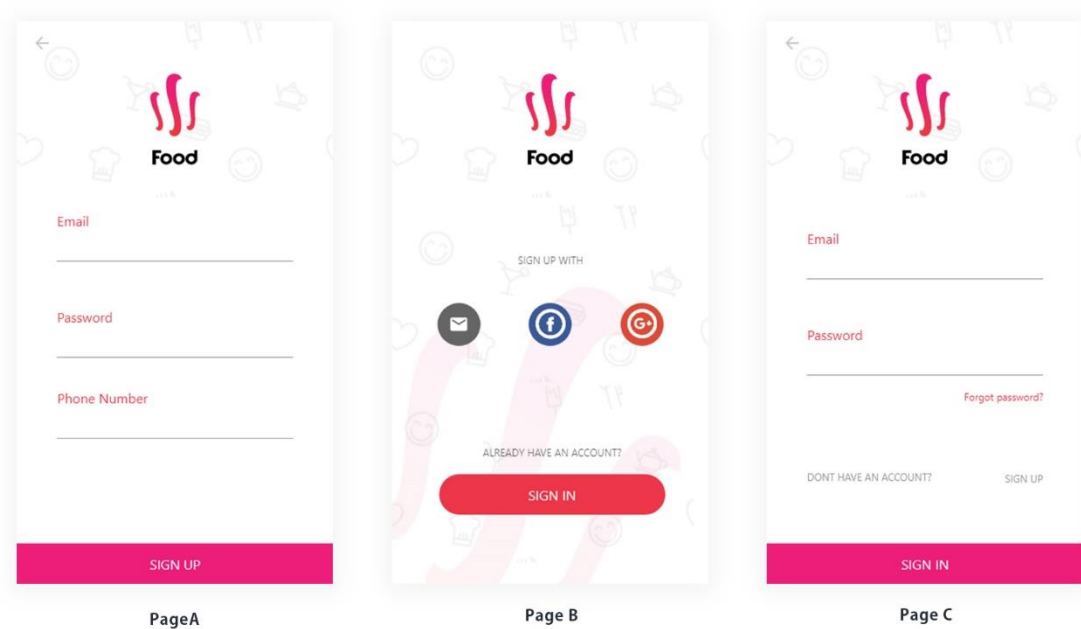


Figure 4.1 - Authentication Pages

The authentication user interfaces are designed to direct the user register or sign into the mobile application with ease. In Figure 4.1, Page A of the authentication pages caters for the registration of a new user. Page B gives a variety of options the user can choose from when registering. Among the available options include registration by email and with third party authentication portals: Facebook and Google for an easier and quicker option for the users.

4.5.2 Home Page for the Customer Application

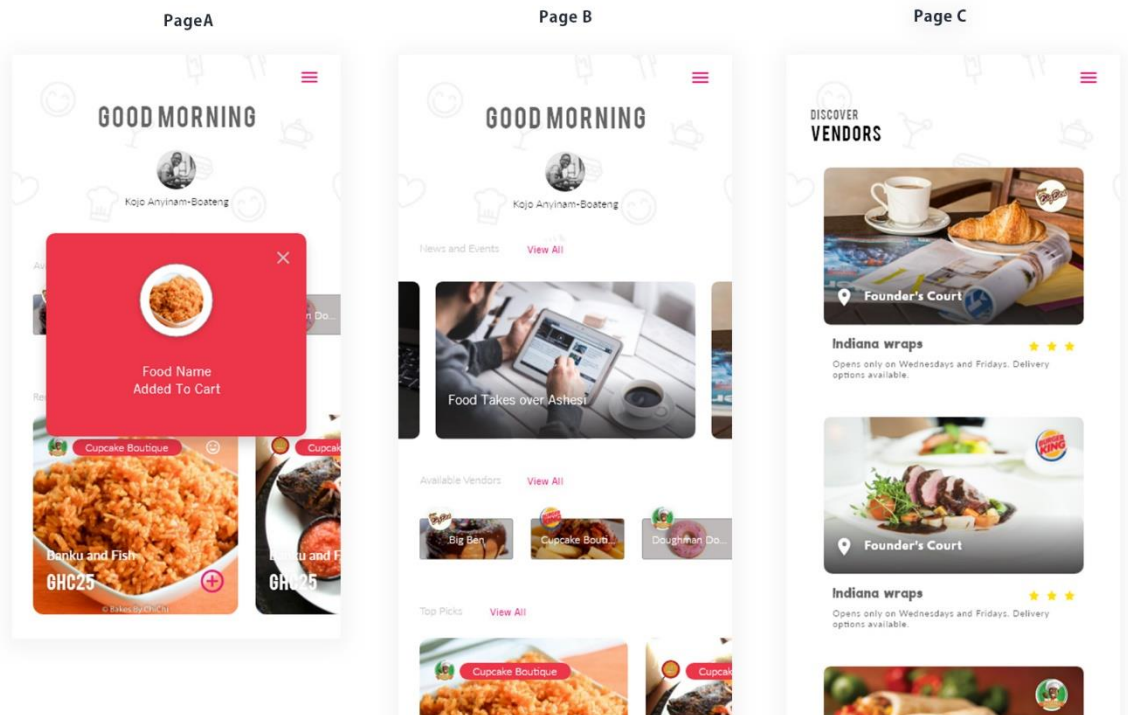


Figure 4.2 – Home Page for Customer Application

After the user has successfully logged into the customer side of the application, the user is greeted based on the time of the day and given the option to select from the variety of available vendors, recommended meals and news articles of the day. In figure 4.2, Page A of the application shows the pop up that appears after the adds one of the recommended meals to cart. Page B shows the homepage of the application when the user logs in. Page C shows a vivid and clearer list of all available vendors when the user clicks on the “View All” button to discover more vendors.

4.5.3 Customer App – Food Store

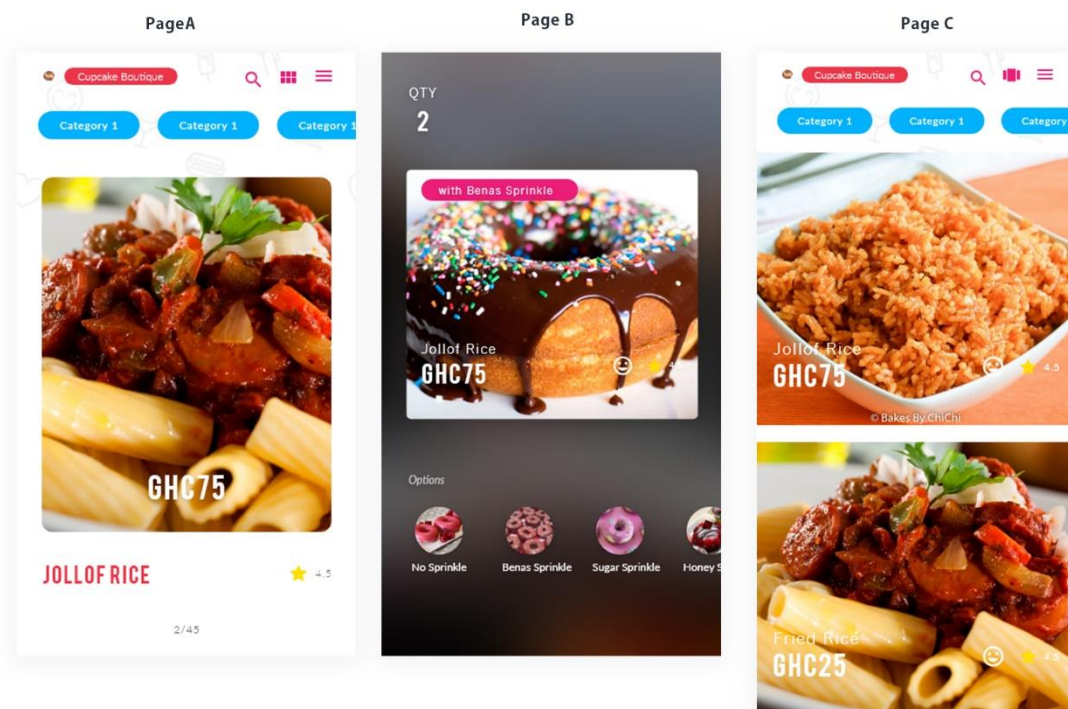


Figure 4.3 – User Interfaces for The Food Store of a Selected Vendor.

The user is met with this page once a vendor from the homepage is selected. In figure 4.3, Page A displays a full card of an item by default with the name of the item, price and rating. The user can then swipe left to view more products available in the store or change the view to that of Page C. Page C shows a vertical display of the products in the store and works with infinite scrolling. Infinite scrolling is a feature that allows users to scroll continually if there exist more items in the list. The items are loaded once the user reaches the end of the page thus allowing more scrolling. Page B shows the page displayed when a user selects an item on the products page. Here the user can increase the quantity by tapping on the right side of the item card or reduce the quantity by tapping on the left side of the food card. The item options represented by circular thumbnails below the main item card gives users the chance to add more options to their selected item.

4.5.4 Vendor Rating Page

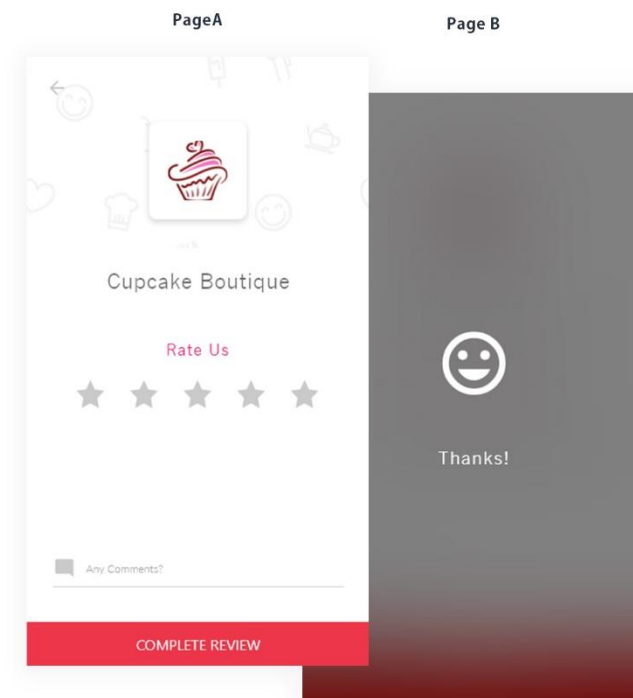


Figure 4.4 – Rating Interface for A Selected Vendor

The rating interface in figure 4.4, was designed and developed to allow users rate their preferred vendors based on the experience they have with them. The rating is done by selecting a preferred number of stars with one star being the lowest and 5 being the highest. The user can also add comments in the comment section below and proceed to complete the review. Once the review is completed, an animation is initiated to appreciate the user's review.

4.5.5 Cart Page

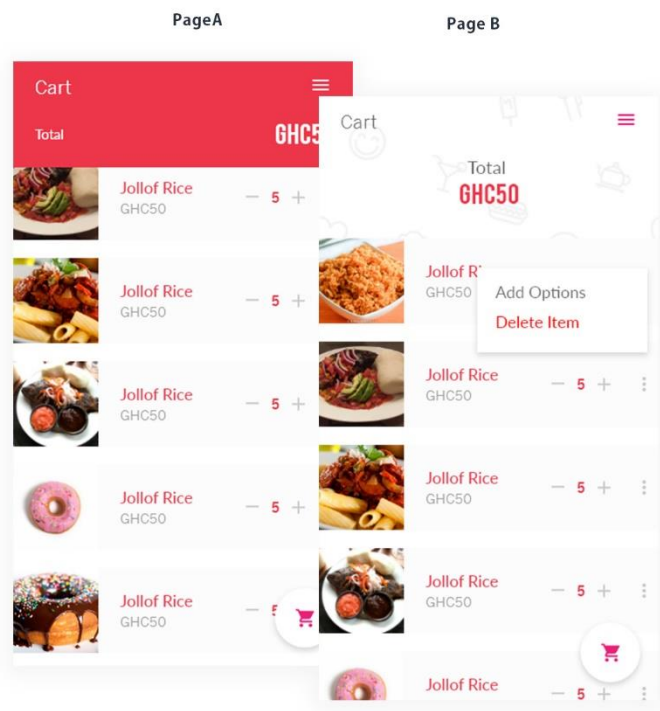


Figure 4.5 – Cart UI

The cart user interface in figure 4.5, handles all selected products ready to be checked out. Users can increase the quantity of an item as remove an item from the list. The user can then checkout using the button below the page with the cart icon.

4.5.6 Checkout Page

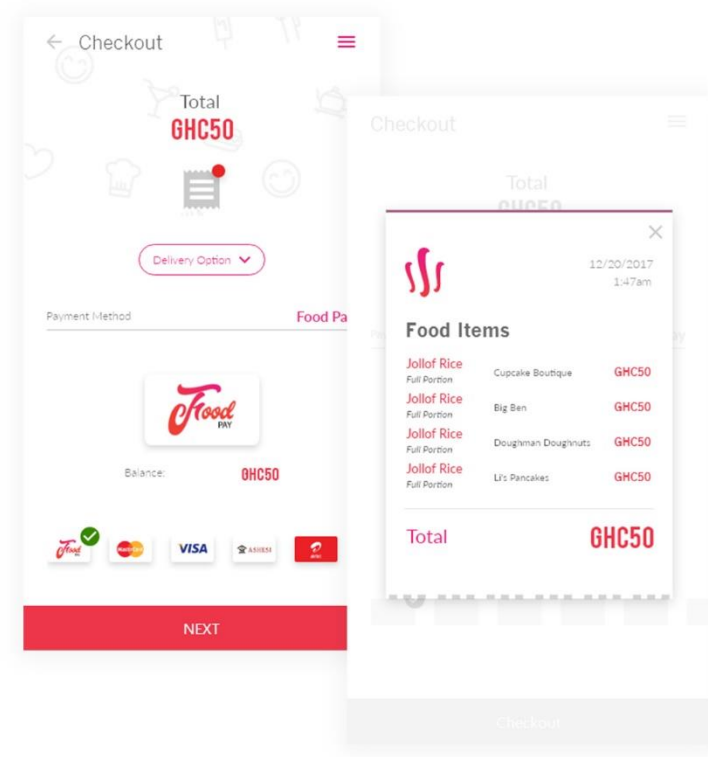


Figure 4.6 – Checkout Page User Interface

The checkout page in figure 4.6, contains a few features. Users can go over their invoice again to view the items they are about to order. This is represented by ticket icon on the page. Another feature is the selection of a preferred delivery option and the third a preferred payment option. Once all these are set up, the user can proceed to make payment.

4.5.7 Delivery Request Page

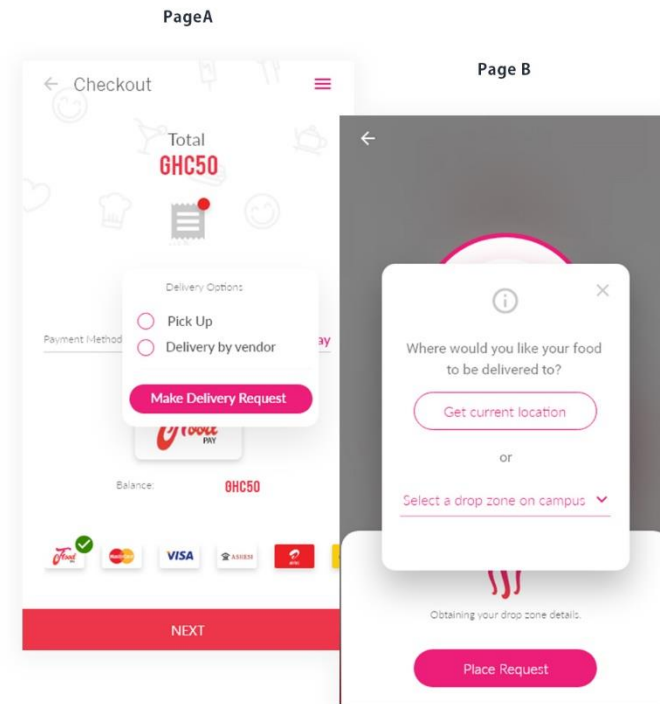


Figure 4.7 – Delivery Initiation Pages

The delivery pages in figure 4.7, are accessed upon selecting the “Make Delivery Request” in the delivery options drop down on the checkout page. Once its selected, the user is directed to the delivery initial page from where his/her location is obtained upon request. Once the location is obtained, the user’s request is broadcasted to other active users of the customer mobile application with a “delivery user” status. The user’s location is obtained and used to find other users off the application within that geofence with the user’s currently obtained location as the center.

4.5.8 Delivery Request Acceptance Page

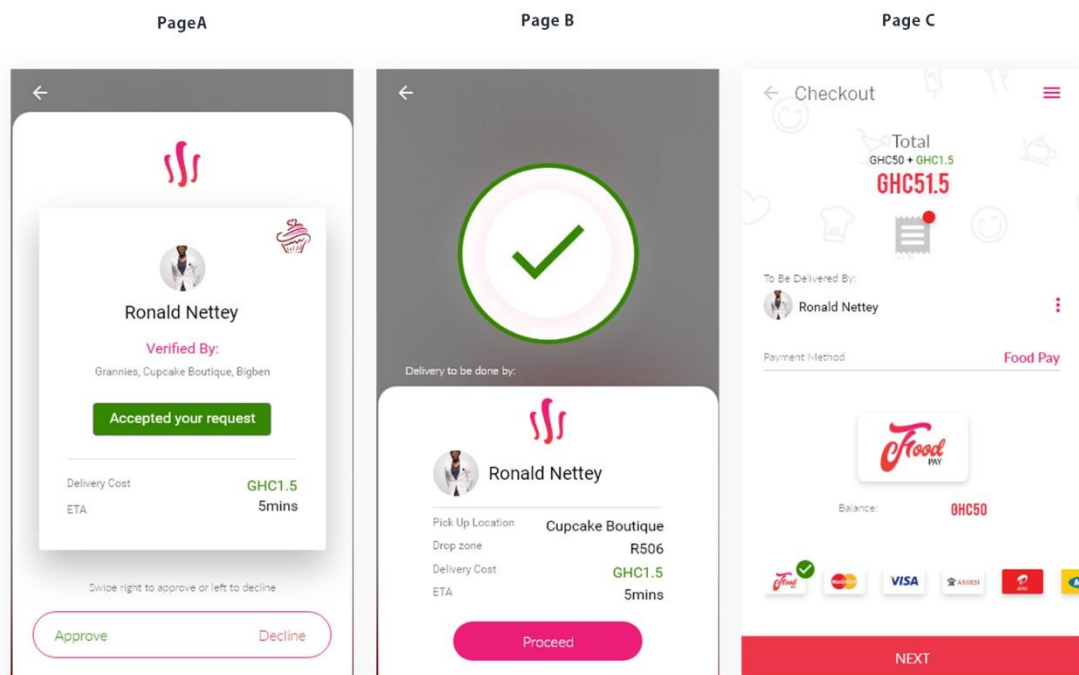


Figure 4.8 – Delivery User Confirmation User Interface

Once another user accepts a customer's delivery request, Page A in figure 4.8, appears. This page gives the user the option to accept or decline the currently present delivery user based on factors which include the delivery cost or the estimated time of arrival. Once the user declines a delivery user, the delivery process request can be reinitialized, and the same steps taken. Upon acceptance of the delivery user, the customer is directed to Page B which displays a summary of the delivery information. The final stage is an update to the checkout page with an addition to the total cost and an avatar display of the delivery user.

Chapter 5: Testing and Result

The main purpose of this chapter is to discuss the test procedures carried to full test the mobile application, the results obtained as well as the feedback received from users of the application. The primary testing procedure which was carried out was the user testing and the secondary being the unit testing. Overall, the applications faired quite well with a few glitches during the testing which will be discussed further in this chapter.

5.1 Unit Testing

Test driven development was not used in the development of this application. However, using the agile software development model, the application was developed in blocks making it easier to identify errors as the project progressed. In addition, throw away prototypes were used a lot before proceeding to the development of the full application thus reducing the margin errors in the complete mobile application.

5.1.1 Block 1 – Fetching of Data from The Cloud Storage and Populating the Products Page.

In the first build of the application, the firebase database which was being used by the Ionic framework was deprecated thus breaking the code completely and causing the application to crash. This was fixed by updating the angularfire2 dependency to the latest versions and updating the modules as well as the syntax of the code. This solved the cloud database access globally in the application as it solely depended on the Google firebase. The database was switched from the real-time Firebase to the Cloud Firestore, a new version of Firebase.

5.1.2 Block 2 – Gradle Update

Subsequent builds of the application yielded not breakages or bugs. However, a major update in Gradle, caused a total crash of not just the application but mainly the Ionic framework. Since the project is dependent on the Ionic framework, massive debugging was done to rectify the problem. However, the break in code persisted and was finally rectified by the Ionic Team. The entire framework required an update from v.3.19 to v.3.20 which cleaned out the errors and improved the compatibility between the Ionic Framework and Gradle version 4.6

5.1.3 Block 3 – Ionic Native Plugins Inclusion.

At this point of development, previous builds have compiled and are working properly. In addition to the current stable builds, native features such as geolocation, network status, unique device identification, local storage and vibrations are added one at a time and tested before proceeding with the development. Plugins which cause a break in the code are immediately discarded and replaced by other open source plugin wrappers. The application runs efficiently on a Samsung J5 Prime running Android Nougat with an API version 7. However, on a Samsung S8, there is a distortion in the layout of elements due to the deprecation of the Cordova plugin, Cross View in Android versions greater than version 8.

5.2 System Testing

After all blocks have been successfully developed and tested, they were combined to form the totality of the application. No prior errors existed, and the system was able to execute tasks as developed to perform. Accessibility of data from the cloud works efficiently with the real time database enabled to persist data for offline functioning of the application. Although the system works correctly when all blocks have been merged, there were a few glitches. There exists some lagging of animations especially at the delivery page of the application which

perform the highest amount of animations which were created using CSS. On higher resolution devices, the animations execute seamlessly however, on lower resolutions, the lagging of animations occur.

5.3 User Testing and Feedback

Deployed on an Android device with an OS version 7 (API level 25), the mobile application was tested using the hall way testing method. Users were given the option to test both the merchant and customer mobile application or one of the two as they preferred. Users who run food businesses on campus were the main testers of the vendor version of the application and users who patronize products from these vendors were the main testers of the customer version of the application. However, no restriction was made on what application a user wanted to test. In addition to that, a form was sent out to a closed group of the Ashesi University community which comprised mainly of the class groups of which only 20 students were selected randomly to test the mobile application. Feedback from these users where sent via email. However, feedback from users during the hallway testing was recorded immediately.

5.3.1 Feedback.

User were unguided during testing obtain accurate feedback after testing. Users were very pleased and appreciated the aesthetics of the application. Most users gave great comments about the layout, color scheme and flow of the customer version of the mobile application. However, out of the total number of testers, 45% of them were uncomfortable with the horizontal sliding of the available vendors and products as illustrated in Figure 4.2 Page B. The reason given was their acclimatization to vertical scrolling. The “View All” option was added to solve that problem and give users the option to scroll through the entire list of vendors both horizontally on the homepage and vertically on the “Discover Vendors” page.

Users also expressed their discomfort in navigation of the application. The feedback obtained was, there included many steps before successfully completing an order on the customer version of the mobile application. Correspondingly, similar feedback was given by users of the vendor version of the Food mobile application. It took too many steps to add new products to the store.

Overall, the application was rated 4.0 of 5 for the customer version of the mobile application and 3.5 out of 5 for the vendor's version of the application. A major concern raised across both platforms was the constant restarting of the application when suspended or minimized. Users found it uncomfortable as there had to repeat navigation steps to reach where they had left off.

Upon analysis of the feedback, the Ionic background mode native plugin had not been added to the applications thus causing them to restart after suspension. The solution thus is to include in the subsequent builds of both applications to curb this problem. The user interfaces as well will be coupled with guidance animations for every page visited for the first time in subsequent builds to reduce the amount of time spent navigating the applications.

Chapter 6: Conclusion and Recommendation

The main purpose of this chapter is to summarize the entirety of the project. It outlines the functional requirements that were met, the limitations and hurdles encountered during the project and the prospective work to be done.

6.1 Summary

The purpose and significance of this project is to develop a food vending, ordering and delivery ecosystem that takes away the inconveniences customers and merchants face when ordering meals and improving customer patronage of food products respectively. The core functionalities which includes, viewing available vendors, purchasing food items, accessing the menu of vendors and sending delivery requests to users were developed but not completely due to the incomplete development of features such as real-time notifications and geofencing. The incompleteness was because of the amount of time spent developing various user interface elements of both mobile applications to suite the needs of the users.

6.2 Limitations and Challenges

Payment methods could not be added to the applications because of the restrictions placed on the payment methods API by Hubtel. It required that the users of the API be verified as a registered company to access the full privileges of the API. Without such privileges, money transferred to the Hubtel merchant account via mobile money could not be transferred out of the account until it had been verified. Other options such as Slydepay also operated with a similar protocol. This posed a major challenge in the implementation of mobile payments thus was taken out of this project and would be implemented in future works due to it being the least preferred mode of payment by users. This information was supported by the results from a questionnaire sent out to the Ashesi community.

The most preferred mode of payment was the meal plan card as 73% of 39 random individuals who answered the questionnaire opted for the meal plan card as their preferred mode of payment. However, at the time of development of this project, the meal plan system was running locally on the Ashesi University network which was a major limitation for the application since user off campus would also be using the application. A request was made for an API to be developed however, the time scope of the project could not allow for its implementation. As a result, the only option for payment is cash, the next preferred payment option of 18.9% of the 39 random individuals who answered the questionnaire.

Another challenge faced during the development of this project was the poorly constructed map of the Ashesi University campus. Detailed data such as roads of specific areas of the campus could not be obtained thus making it difficult to include the real time directions using Google Maps. It often generated a straight line from one point to another when the user used a route which had no detailed directions on the map.

6.3 Future Work

- i. Development for IOS devices:

The application was only developed for Android devices due to the lack of resources which were needed for IOS development. Such resources include a MacBook, an iPhone and an Apple developer account. Such resources would be acquired later to develop the application for IOS devices.

- ii. Geofencing:

The customer's application is populated with vendors created in Ashesi. Should the user leave campus, there is no restriction to update the vendor list on the platform thus menus of vendors can be accessed out of Ashesi which breaches the peer to peer protocol of the application which allows peers closer to the user making the delivery request receive delivery request notification. If the distance between the customer and the vendor is very large, it creates an inconvenient experience which violates the while purpose of the application. As a way of preventing this geofencing could be used. When a user enters a geofence, the menu of the vendors within that radius is displayed on the application. It is updated upon leaving and entering other geofences.

iii. Multiple payment option inclusions:

Unfortunately, the only mode of payment is cash. However, once the requested meal plan API has been fully tested, it will be integrated with the application. The goal is also to obtain business certification documents which can be used to gain access to other payment portal to create a wide variety of payment option for the users of the application.

6.4 Conclusion

Paying attention to the needs of the users, an aesthetically pleasing and intuitive application has been developed to improve customer experience, delivery and management of food orders and inventory efficiently. Although it exists as an application developed for the Ashesi University, plans are underway to make this product market ready. Students, staff and faculty can now order their preferred meal, and have it delivered to them by a fellow member of the Ashesi community. New friends can be made. Small food businesses can also reach out and interact with the community via in app messaging and notifications. Students can also take up

delivery roles only to accumulate some revenue. Individuals can now make use of the time spent in queues to complete other pending tasks. This project does not just ease the community of the challenges and inconveniences faced with getting a meal for the day but also creates opportunities for everyone.

References

- Adobe. (2018). *Introducing Adobe XD*. Retrieved from Adobe: <https://www.adobe.com/africa/products/xd.html>
- Adobe. (2018). *The state of the art of illustration*. Retrieved from Adobe: <https://www.adobe.com/africa/products/illustrator.html>
- Angular. (2018). *One Framework. Mobile and desktop*. Retrieved from Angular: <https://angular.io/>
- College, A. U. (2014, September 16). *The anatomy of Ashesi's Class of 2018* . Retrieved from Ashesi : <http://www.ashesi.edu.gh/stories-and-events/1880-the-anatomy-of-ashesi-s-class-of-2018.html>
- College, A. U. (2017, September 10). *Welcoming the Class of 2021*. Retrieved from Ashesi : <http://www.ashesi.edu.gh/stories-and-events/2519-ashesi-welcomes-class-of-2021.html>
- Elliott, R. (2015, March 9). *Mobile usage and popular brands in Africa*. Retrieved from GeoPoll: <https://blog.geopoll.com/mobile-usage-africa>
- Google. (2018). *Firebase*. Retrieved from Firebase: <https://firebase.google.com/>
- Hirschberg, C., Rajko, A., Schumacher, T., & Wrulich, M. (2016, November). *The Changing Market for Food Delivery*. Retrieved from McKinsey & Company High Tech: <https://www.mckinsey.com/industries/high-tech/our-insights/the-changing-market-for-food-delivery#0>

Ionic. (2018). *Build amazing apps in one codebase, for any platform, with the web.* . Retrieved from Ionic Framework: <https://ionicframework.com/>

[1] *JUMIA ANNUAL MOBILE REPORT 2018 LAUNCHED IN ACCRA.* (2017, April). Retrieved April 14, 2018, from Government of Ghana: <http://www.ghana.gov.gh/index.php/media-center/news/4470-jumia-annual-mobile-report-2018-launched-in-accra>

Liu, Y., Guo, B., Du, H., Yu, Z., Zhang, D., & Chen, C. (2017). Poster: FooDNet. *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking - MobiCom '17*(August), 564-566. Retrieved from <http://dl.acm.org/citation.cfm?doid=3117811.3131268>

Mozilla. (2018). *HTML5*. Retrieved from MDN Web Docs: <https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5>

Othman, Z., Salehuddin, M., Zahari, M., & Jalis, M. H. (2009). Restaurant Service Delivery Attributes and Customer Patronization : Comparison of Two Malaysian Ethnic Restaurants. 67-82.

SASS. (2018). *CSS with superpowers* . Retrieved from SASS: <https://sass-lang.com/>

Schumacher, T., & Swinford, D. (2016, June). *How a tech unicorn creates value*. Retrieved from McKinsey & Company: <https://www.mckinsey.com/business-functions/strategy-and-corporate-finance/our-insights/how-a-tech-unicorn-creates-value>

Shneiderman, B. (2016). Designing the User Interface: Strategies for Effective Human-Computer Interaction: Sixth Edition. In B. Shneiderman, *The Eight Golden Rules of Interface Design*. Pearson.

Statista. (2017, April). *Annual revenue of Delivery Hero worldwide from 2013 to 2016 (in million euros)*. Retrieved from Statista: <https://www.statista.com/statistics/696451/delivery-hero-revenue/>

Syed Ghayas, S. S. (2013). The Effects of Icon Characteristics on Users' Perception. *Advances in Visual Informatics* , 652-663.

Technavio. (2017, November). *Global Online On-Demand Food Delivery Services Market 2017-2021*. Retrieved from Technavio: https://www.technavio.com/report/global-online-on-demand-food-delivery-services-market?utm_source=t6discount&utm_medium=bw&utm_campaign=businesswire

TypeScript. (2018). *Typescript. Javascript that scales*. Retrieved from Typescript: <https://www.typescriptlang.org/>

Wire, B. (2017, December 13). *Online On-demand Food Delivery Services Market - Growth Analysis and Forecast/ Technavio* . Retrieved from Business Wire: <https://www.businesswire.com/news/home/20171213006080/en/Online-On-demand-Food-Delivery-Services-Market-->