CMPE 492

Senior Design Project II

Final Report

Project Name: Cakery App

Project Members:

Selin Akyurt - 17878041578

Melisa Uzulu - 3906408574

Recep Berkay Engin - 22271682594

Supervisor: Yücel ÇİMTAY

Jury Members: Aslı Gençtav, Venera Adanova

1. INTRODUCTION

1.1 Motivation and Background

Cakery is here to help you with the problem of visiting bakeries and buy/order cakes for your celebrations and it will save your precious time making bakeries available for you with just few clicks.

Ordering a cake for a special occasion from a bakery can be time consuming and stressful task, especially with our busy schedules. Customized cakes, such as those for birthdays and engagement/wedding parties, require a lengthy process of visiting multiple bakeries, discussing preferences, comparing prices and more. That's where our project application "Cakery" comes in.

With Cakery, you can easily access a list of local bakeries, view their products with detailed descriptions and place orders for customized or ready to eat cakes, all with few clicks with the comfort of your phone. Bakeries will then deliver the order to your desired location on the desired date and time. This way we will eliminate the time-consuming trips and consultations for you.

Cakery is also a platform where small businesses can showcase their unique products and reach a wider audience. With our app Cakery, we hope to give them the recognition they deserve for their hard work, while making the order process easier and more efficient for everyone involved.

The project is consisted of 3 different systems that requires careful planning and integration. Throughout the semester we are aiming to successfully put together a seller app, user app and an admin web portal which forms the Cakery using Flutter as a framework and Dart as a programming language. After successfully create our system basis, we will continue with testing the prototypes and the whole system together, add any necessary adjustments and improve the system to make the user experience easier. For the last step, we are hoping to extend the Cakery App for commercial use.

1.2 Impact Of The Solution

A) Environmental Context

The traditional way of ordering a cake from a bakery for our special days is going through few bakeries, examining the cakes they have made so far, show the cake you want them to bake, get prices from each bakery and compare the prices to choose the right bakery. This is a time-consuming and stressful task. Visiting each bakery individually by car or public transportation causes more exhaust emissions to be released into the atmosphere. With Cakery, we resolve the problem of distributing appropriate cake designs to appropriate customer base. This centralized structure of Cakery system helps to reduce potential logistical waste issues.

B) Global Context

Ordering a customized cake from a bakery process is a mechanical problem. It includes, visiting bakeries one by one, discussing your personal preferences and comparing prices. In the best case, even if the bakeries have an online presence on the Internet, a customer needs to search for the bakeries one by one which is yet again time consuming. Cakery solves this all with few clicks with the comfort of your phone. Secondly, Cakery offers its member bakeries the opportunity to reach a much larger customer base and cater to a broader customer segment.

C) Economic Context

Global pandemic Covid-19 affected buying behaviour in a way that a huge number of consumers shifted their consuming habits from traditional paying to online paying. More people started ordering online for time-saving and health concerns. This shift in consuming habits made a domino effect in the market for online food delivery. Also, customized products are becoming more and more in demand from consumers, including in the food sector. Businesses that sell bespoke cakes, like Cakery have a great chance because to this. Lastly, Cakery gives a flexible pricing opportunity for bakeries, allowing them to charge the custom cakes. This creates a competitive pricing which is an economic advantage in attracting wider range of customer base.

D)Societal Context

Cake consumption as a tradition in society provides a steady stream of customers for Cakery. This includes birthdays, anniversaries, weddings, and other noteworthy occasions. Also, Cakery supports local business by giving a chance for local bakeries to have an online presence on the same dashboard with global bakeries.

1.3 Contemporary Issues

Cakery's main contemporary issue is privacy since our system is highly dependent on user data. The system provides end-to-end encryption as an essential feature. Also, Cakery application is managed by an admin. In the admin interface, a user or bakery account can be activated or blocked if any suspicious action is reported.

1.4 New Tools and Technologies

Flutter 3: Flutter is an open-source UI software development kit created by Google. It is used to develop applications for Android, iOS, Windows, Mac, Linux and the web.

Dart: Flutter runs on dart, a programming language developed by Google. Dart is a client optimized language for developing fast applications on any platform. Its aim is to offer the most productive programming language for multiplatform development paired with a flexible execution runtime platform for application frameworks.

Cloud Firestore: Cloud Firestore is a cloud-hosted NoSQL database that your iOS, Android and web apps can access directly via native SDKs. Query the data as you wish and it is structured. Its data is synchronized between devices online or offline.

Git/Github: Git is a version control system for tracking version of computer files and coordinates coordinate team-working by tracking files among multiple people. Github is a web-based hosting service for version control via Git.

Android Studio: The official Integrated Development Environment (IDE) for Google's Android operating system, built on IntelliJ IDEA from JetBrains, is known as Android Studio. It works smoothly with the built application because of its built-in android emulators.

Visual Studio Code: Microsoft created the source-code editor Visual Studio Code, generally known as VS Code, for Windows, Linux, and macOS using the Electron Framework. Debugging support, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git are among the features.

1.5 Use Of Libraries and Resources

Shared_preferences: Flutter plugin for reading and writing simple key-value pairs. Wraps NSUserDefaults on iOS and SharedPreferences on Android.

Image_picker: Flutter plugin for selecting images from the Android and iOS image library and taking new pictures with the camera.

Cupertino: Flutter widgets implementing the current iOS design language.

Firebase_core: Flutter plugin for Firebase Core, enabling connecting to multiple Firebase apps.

Firebase_auth: Flutter plugin for Firebase Auth, enabling Android and iOS authentication using passwords, phone numbers and identity providers like Google, Facebook and Twitter.

Firebase_storage: Flutter plugin for Firebase Cloud Storage, a powerful, simple, and cost-effective object storage service for Android and iOS.

Cloud_firestore: Flutter plugin for Cloud Firestore, a cloud-hosted, noSQL database with live synchronization and offline support on Android and iOS.

Geolocator: Geolocation plugin for Flutter. This plugin provides a cross-platform (iOS, Android) API for generic location (GPS etc.) functions.

Fluttertoast: Toast Library for Flutter, easily create toast messages in single line of code.

1.6 System Requirements

Number	Functional Requirement Explanation	Admin Web Portal (A)
Req-1	Admin will enter their e-mail.	*1. Login Page Module*
Req-2	Admin will enter their password.	
Req-3	Admin interacts with login button to log in the system.	
Req-4	If the e-mail is invalid, admin gets error message.	
Req-5	If the password is invalid, admin gets error message	

Req-7	If the e-mail is already taken the admin gets an error message.	*2. Register Page Module*
Req-8	If the admin leaves the password blank, an error message will appear	
Req-9	Admin fills in the name and surname information in the required field.	
Req-10	Admin fills in the email information in the required field.	
Req-11	Admin fills in the password information in the required field.	
Req-12	If admin enters an incorrect e-mail address, admin will receive an error message.	
Req-13	When all the necessary information is filled, the information is saved and transferred to the main page by clicking on the save part	
Req-14	If there is any missing information, it will encounter an error message.	
Req-15	When the admin wants to change the password, admin enters the password in the password field and confirms it in the confirm old password to save field.	
Req-16	If the newly entered password meets the required password qualifications, the confirmation will take place	
Req-17	If it does not meet the required password qualifications, it will get an error message.	
Req-18	Admin can view active user accounts.	*3. Admin Portal Page Module*
Req-19	Admin can view active seller accounts.	
Req-20	Admin can view block user accounts.	
Req-21	Admin can view block seller accounts.	
Req-22	Admin can see the date, time, and month information on the screen.	
Req-23	If admin wants to exit the portal page, admin clicks the logout button.	
Req-24	Admin can click to view active user accounts.	*3.1 Active User Accounts Page Module*
Req-25	Admin can view the user's name-surname.	
Req-26	Admin can see the user's phone number.	
Req-27	Admin can view user's photo.	
Req-28	Admin can click "Block this account" button to remove the user from the system.	

Req-29	If the admin blocks the user, the user is displayed on the blocked user’s page.	
Req-30	Admin can click to view block user accounts.	*3.2 Block User Accounts Page Module*
Req-31	Admin can view the block user's name-surname.	
Req-32	Admin can see the block user's phone number.	
Req-33	Admin can view user's photo.	
Req-34	If the admin wants to add the user back to the system, they can click the "Activate this account" button.	
Req-35	The user that the admin reactivated is displayed in the "active user accounts" section.	
Req-36	Admin can click to view active seller accounts.	*3.3 Active Seller Accounts Page Module*
Req-37	Admin can view the company names of the active sellers.	
Req-38	Admin can see the active seller's phone number.	
Req-39	Admin can view the seller's photo.	
Req-40	Admin can click "Block this seller" button to remove the seller from the system.	
Req-41	If the admin blocks the seller, the user is displayed on the blocked seller’s page.	
Req-42	Admin can click to view block seller accounts.	*3.4 Block Seller Accounts Page Module*
Req-43	Admin can view the company names of the block sellers.	
Req-44	Admin can see the block seller's phone number.	
Req-45	Admin can view block seller's photo.	
Req-46	If the admin wants to add the seller back to the system, they can click the "Activate this seller" button.	
Req-47	The seller that the admin reactivated is displayed in the "active seller accounts" section.	

Number	Functional Requirement Explanation	Cakery App (B)
Req-48	User will enter their e-mail.	*4. User Login Page Module*
Req-49	User will enter their password.	
Req-50	User interacts with login button to log in the system.	
Req-51	If the e-mail is invalid, user gets error message.	
Req-52	If the password is invalid, user gets error message	
Req-53	User will click on the "Register" page, if they don't have an account.	

Req-54	If the e-mail is already taken the user gets an error message.	*5. User Register Page Module*
Req-55	If the user leaves the password blank, an error message will appear.	
Req-56	User fills in the name and surname information in the required field.	
Req-57	User fills in the email information in the required field.	
Req-58	User fills in the password information in the required field.	
Req-59	User fills in the confirm password information in the required field.	
Req-60	If user enters an incorrect e-mail address, user will receive an error message.	
Req-61	User can upload photos to required field.	
Req-62	When all the necessary information is filled, the information is saved and transferred to the main page by clicking on the save part.	
Req-63	If there is any missing information, it will encounter an error message.	
Req-64	User can see the current profile information in the input boxes	*6. User Profile*
Req-65	User can see the name-surname.	
Req-66	User can see the current profile photo.	
Req-67	User can change their own address.	*6.1. Edit User Address*
Req-68	The user can see the "information" section where user can view her/his own information.	*7. User Dashboard Module*
Req-69	User can view featured sellers.	
Req-70	User can view all sellers.	
Req-71	Users can view menu when they click on the sellers.	
Req-72	Users can see their name and surname.	*7.1 User Information Module*
Req-73	Users can view the current order with the "my orders" button.	
Req-74	Users can view the past order with the "history" button.	
Req-75	Users can search for vendors or products with the search button.	
Req-76	Users can "add new address" new addresses.	
Req-77	Users can exit the system with the logout button.	
Req-78	User fills in the name and surname part appropriately.	

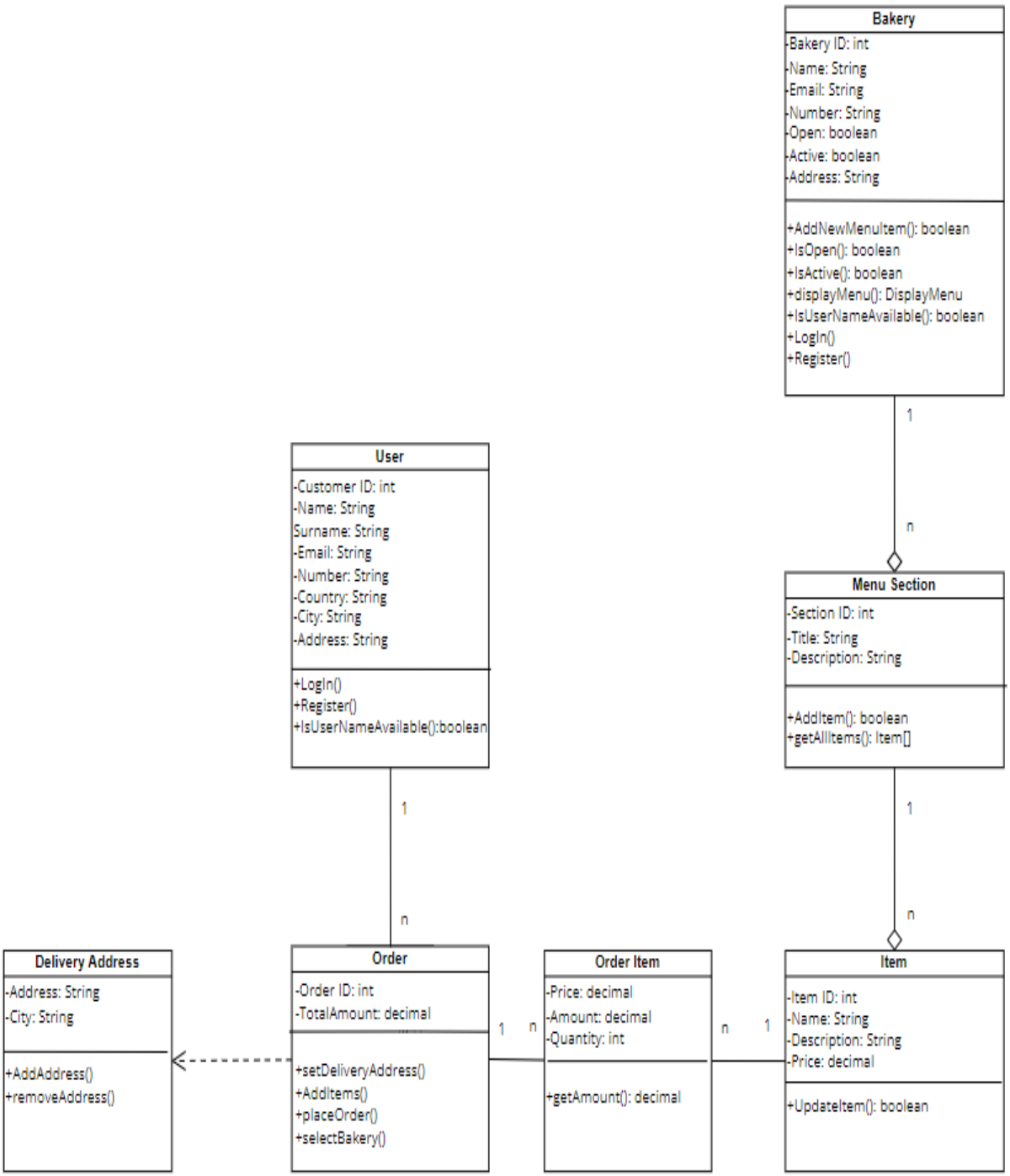
Req-79	User fills in the phone number section appropriately.	*7.1.1 Add New Address Module*
Req-80	User fills in the address part appropriately.	
Req-81	User fills in the city part accordingly.	
Req-82	User fills in the country part appropriately.	
Req-83	User fills in the full address part accordingly.	
Req-84	If the user does not know his exact address, user can add his/her current location to the automatic system with the "Get My Current Location" button.	
Req-85	The user can see the menu of a seller she/he wants.	*7.2 Menu View Module*
Req-86	Users can view all products in the menu.	
Req-87	Users can see the prices of the products.	
Req-88	Users can view information about products.	
Req-89	Users can select the number of products to be added to the cart.	
Req-90	Users can click the "add to cart" button to add the product to the cart.	
Req-91	Users can see the photo of the product.	
Req-92	Users can see how many items are in the cart.	
Req-93	Users can view the name of the product.	
Req-94	If the user wants to make a customized product, he/she can click the customized cake option.	
Req-95	By clicking on the customize cake option, the customer goes to the customization page of the personalized product.	*7.2.1 Customized Menu Module*
Req-96	The user can upload a picture of the product on this page.	
Req-97	The user can write an information note about the content of the product.	
Req-98	The user can see the portion types of the product and can choose between portion types.	
Req-99	The user can choose from the materials to be added to the product as he/she wishes.	
Req-100	The user can choose from the delivery options of the product.	
Req-101	When the user adds more products, they can see the updated price of the product.	
Req-102	The user can send information to the seller by clicking the "Send Cake for Approval" button.	

Req-103	The user can press the back button to view another product.	
Req-104	The user can send the product he/she has prepared to the seller to get the price of the product.	
Req-105	Users will be able to view all the products added to the cart here.	*7.3 Cart Page Module*
Req-106	Users will be able to see the name of the product.	
Req-107	Users will be able to see the quantity of the product.	
Req-108	Users will be able to see the price of the product.	
Req-109	Users will be able to click on the "Clear Cart" button if they want to remove all the products in the cart.	
Req-110	Users will be able to see the total cart amount.	
Req-111	Users are directed to the address page to determine the address of the place where they will place an order.	
Req-112	Users will be able to choose their registered address.	*7.3.1 Address Page Module*
Req-113	Users will be able to add a new address if they do not have a registered address.	
Req-114	Users will be able to use their current location addresses instead of their registered home addresses.	
Req-115	After completing the address steps, users will be directed to the page where they can see the order details.	
Req-116	Users will be able to view information about successful order.	*7.3.2 Successful Order Page Module*
Req-117	Users will be able to view the amount of the order.	
Req-118	Users will be able to see the address to which the order will be sent.	
Req-119	Users will be able to see the name and surname of the person who placed the order.	
Req-120	Users will be able to see the order date and time.	
Req-121	Users will be able to see the order ID.	
Req-122	Users will be able to see the phone number.	
Req-123	Bakery will enter their e-mail.	*8- Bakery Login Module*
Req-124	Bakery will enter their password.	
Req-125	Bakery interacts with login button to log in the system.	
Req-126	If the e-mail is invalid, bakery gets error message.	
Req-127	If the password is invalid, bakery gets error message.	
Req-128	Bakery will click on the "Register" button if they don't have an account.	

Req-129	Bakery will enter their e-mail.	
Req-130	If the e-mail is already taken the bakery gets an error message.	*9- Bakery Register Module*
Req-131	If the bakery leaves the password blank, an error message will appear.	
Req-132	Bakery fills in the name information in the required field.	
Req-133	Bakery fills in the email information in the required field.	
Req-134	Bakery fills in the password information in the required field.	
Req-135	Bakery fills in the confirm password information in the required field.	
Req-136	If bakery enters an incorrect e-mail address, bakery will receive an error message.	
Req-137	Bakery can upload photos to required field.	
Req-138	When all the necessary information is filled, the information is saved and transferred to the main page by clicking on the save part.	
Req-139	If there is any missing information, it will encounter an error message.	
Req-140	If it does not meet the required password qualifications, it will get an error message.	
Req-141	There is an "information" section where the baker can see their own information.	*10-Bakery Dashboard Module*
Req-142	Bakery can see its items/menu on listed order.	
Req-143	Bakery can view details when it clicks on an item.	
Req-144	Bakery can see a “Add New Menu” button on right top.	
Req-145	Bakery can see items in descending order.	
Req-146	Bakery can see their name.	*11-Bakery Information Module*
Req-147	Bakery can see their logo picture.	
Req-148	Bakery can view the past order with the "History" button.	
Req-149	Bakery can view the present orders with the "New Orders" button.	
Req-150	Bakery can see its home page with the “Home” button.	
Req-151	Bakers can see customizable cake requests from customers by clicking on the "Custom Cake Approval" button.	
Req-152	Bakery can exit the system with the logout button.	

Req-153	If there's a new order visible, bakery can click on the order to see detailed information of the order.	*11.1-Bakery Orders Module*
Req-154	Bakery can see shipping details of new order.	
Req-155	Bakery can see the price of new order.	
Req-156	Bakery can see history of orders from "History".	
Req-157	Bakery can see amount of an item that has recently sold.	
Req-158	Bakery can see price of an item that has recently sold.	
Req-159	Bakers can see information about customizable cake requests from customers on the "Custom Cake Approval" page.	*11.2 - Bakery Customized Cake Module*
Req-160	Bakers can see request letters from customers.	
Req-161	Bakers can see the portion size selected by customers.	
Req-162	Bakers can see the variety of flavours chosen by customers.	
Req-163	Bakers can see the embellishment option chosen by customers.	
Req-164	Bakers can see the type of order selected by customers.	
Req-165	Bakers can see the quantity of products selected by customers.	
Req-166	There is an area for bakers to specify a price against customers' special cake requests.	
Req-167	Bakers have the option to accept or decline this cake request.	
Req-168	Bakery can create/adjust its Menu.	*12 - Bakery Edit Menu Module*
Req-169	Bakery can create an item for it's menu by clicking on the "Add New Item" button on top right.	*12.1- Bakery Create Item Module*
Req-170	Bakery can upload a picture of the new item.	
Req-171	Bakery can fill in item information.	
Req-172	Bakery can set a price for the new item.	
Req-173	Bakery can delete an item by selecting delete icon that is displayed in detailed screen.	*12.2- Bakery Edit Menu Module*

1.7 Initial Class Diagram



2. RELATED SYSTEMS

For related systems, we are going to display the overview of different technologies and systems that helps Cakery App success and functioning it properly.

- Geolocation Services: Cakery uses the help of Google Maps API to provide accurate geolocation services. Those geolocation services mostly used for the functionality where user/seller allows the application to get their current location to save their address while enhancing the overall user experience. With this help, users can easily fill their address and complete ordering process. For the security purposes we also ask every seller to give their current location based on google maps API while signing into our seller app.

- Google Drive Integration: Through Google Drive integration, users can easily upload, store and manage pictures while using the app. Cakery allows users/sellers to choose pictures from their google drive accounts to make this process easier. Users can use their google drive accounts to upload their profile pictures or place an example picture of their desired custom cake through their custom order process. Sellers can place the pictures of their selling items and menus, as well as uploading profile picture just like users. This provides Cakery users to have the ability of personalizing their profiles and cake orders.

3. PROJECT OVERVIEW:

Cakery App aims to provide convenient, easy to use, user friendly platform to users to order their special day cakes from different range of bakeries as well as enhancing local bakery visibility. Our goal was to make the cake ordering process easier and down it into few clicks with seamless user experience while saving time for our users.

Cakery App can be broken into 3 different subsystems. User app, which allows users to see the bakeries and place cake orders from them. Seller app allows bakeries to sign up to the system and upload their menu items into their bakery which will end up displaying on the user app and take orders from the users. Admin Web Portal is a website where admin is responsible for the whole user/seller app safety and can block users/sellers in a misusage.

For all three subsystems we used Flutter and Dart programming languages. Flutter gave us an opportunity to code cross-platform and came up with a user-friendly interfaces and engaging features. We used Android Studio as our development environment and it helped us to code, debug and test our application.

4. IMPLEMENTATION

First we decided to build it as a mobile app in Flutter. Then we decided to divide our application into 3 parts. These are called Cakery App (For Seller) to be used by Sellers, Cakery

App (For User) to be used by customers, and finally Admin Web portal to be used by admins who will manage the system. Although we did not have the chance to experience the Dart language used by Flutter, we did the necessary research and started developing our project with the help of Flutter after we were ready for the basic structure of Dart. Knowledge of Java, an object-oriented language we used in the past, helped us a lot. Thanks to Flutter and Google Firebase, we easily provided communication between our application and the database. By doing this, we have ensured that anyone who can access our vision app can use the app securely.

Our plan was to create this mobile application during the fall semester. To achieve this, we started by dividing our mobile applications into two parts. At the last stage, we created a web portal page for the admin. We achieved this thanks to the Dart language we learned to develop HTML and Flutter applications.

4.1 Cakery App (For Seller): We started the development phase by first creating the app for the vendors to use. We have created login/register pages for the seller to register and log into the system on the seller's side. Then, we decided what the Sellers would need in their panels and developed pages such as the menu page, the product page, the history page where they can view past orders, and the custom cake approval for custom cakes, where the cake requests of the users to the vendors will be displayed. Thanks to the Google Location Service that we add to our system while the sellers are registering, their location can be determined automatically by the system. In addition, sellers can upload photos via Google drive to the product page. While designing this application, we tried to take advantage of Google Services. On the page where the sellers display their products, there are also fields for information about the contents of the products, information about the price of the product, and the quantity of the product. The products that the sellers add to their menus appear instantly in the users' systems. There is also an exit button to exit the system. Even if the mobile device is turned off, the last login account remains open.

4.2 Cakery App (For User): Secondly, in our development phase, we started to develop the application that users will use. In the application of the users, there are register/login pages for registering and logging into the system. We got help from Google Location services so that users can easily register their addresses while registering to the system. The menus of many different vendors and the products offered by the vendors appear instantly in the users' system. We have developed pages so that users can view their past orders, see the products they have ordered and add new addresses. While making these improvements, we took advantage of the widget structure that Flutter provided to us. We have also developed a custom cake page that users can access from each vendor so that they can request and order different cakes. On this page, there is an image upload section that works with Google Drive and mobile device gallery, where users can upload images of the cakes they want. In addition to these, there are text fields, portion selection, flavor selection section, cake decoration selection section, shipping type selection section and product quantity selection section for users who want to convey their requests in writing. There is also a safe exit button to exit the system. Even if the mobile device is turned off, the last login account remains open.

4.3 Admin Web Portal: We developed a web portal for admin in the last development phase of our project. We tried to develop a system where the admin can view information about their sellers and users. Thanks to this system, we have given the admin the authority to block users or vendors who cause problems. User and seller blocked from the system cannot enter the system until they are reactivated by the admin. We did not neglect to show this warning message on the login screen of blocked Users and sellers.

4.4 Custom Cake Processing

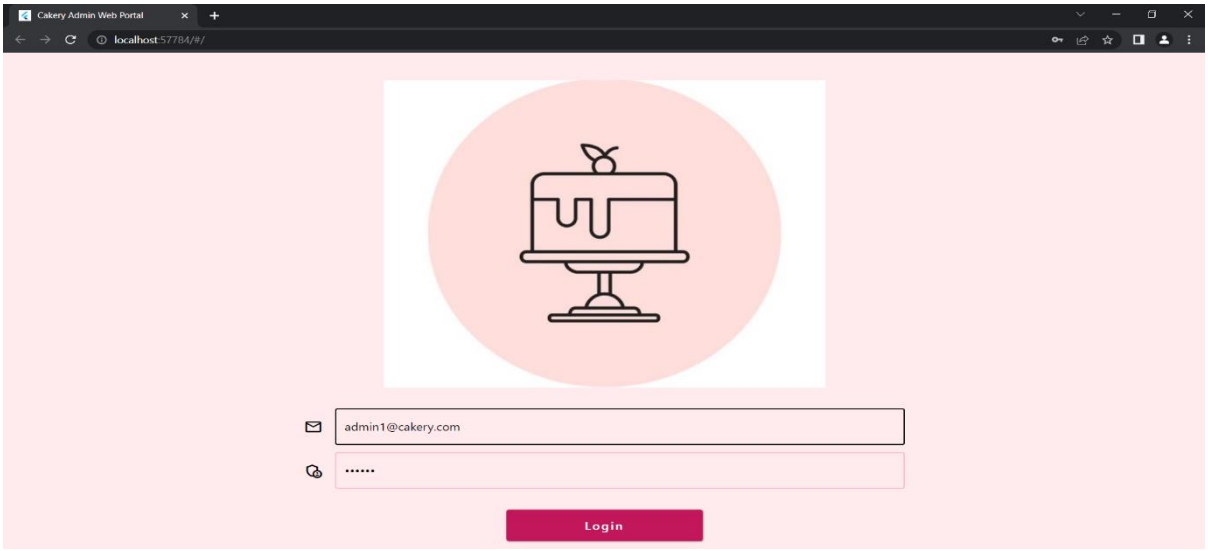
One of the most important features of the Cakery App is the Custom Cake section. Thanks to Custom Cake, we have developed a custom cake page that users can access from any vendor so that they can request and order various types of cakes. On this page, there is an image upload section that works with Google Drive and mobile device gallery, where users can upload images of the cakes they want. In addition to these, there are text fields, portion selection, flavor selection section, cake decoration selection section, shipping type selection section and product quantity selection section for users who want to convey their requests in writing. There is also a safe exit button to exit the system. Custom cake enhancements are available in the code files at https://github.com/melisauzulu/cakery_repo_user, and

https://github.com/melisauzulu/cakery_app . The reason we did this part was to meet the needs of the users easily. How the system works is explained below.

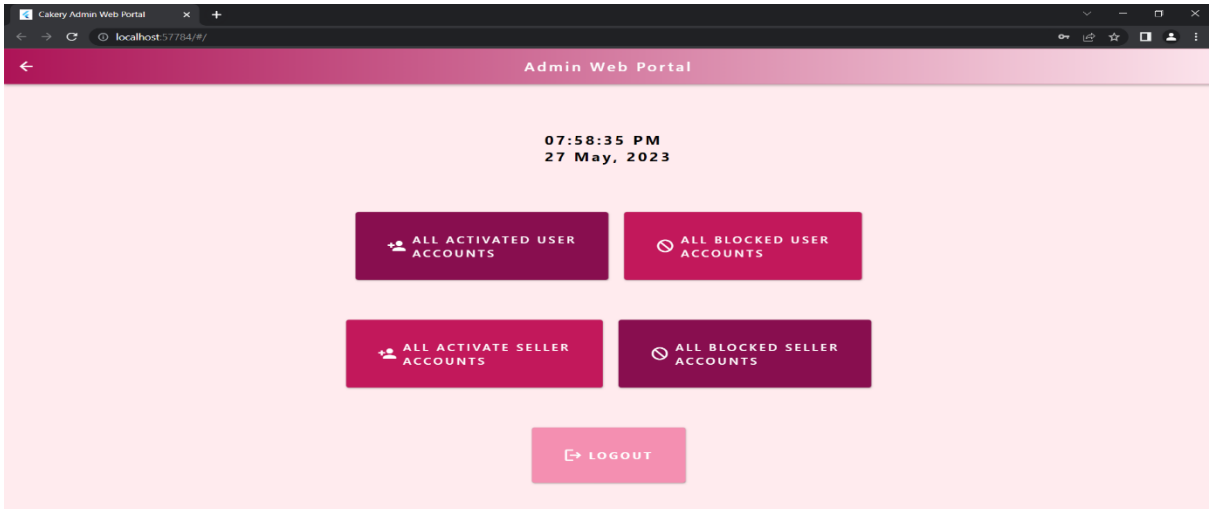
After any user logs into the system, the bakery in the immediate vicinity registered in the system is displayed on the homepage. The user can access the menu of the seller (bakery) he wants. After the user accesses the menu, there is an Add Custom Cake button at the top of the menu. After clicking this button, a new page opens on the user's screen. On this page, there is an image upload section that works in harmony with Google Drive and mobile device gallery, where users can upload images of the cakes they want. In addition to these, there is a portion selection, flavor selection section, cake decoration selection section, shipping type selection section and product quantity selection section, along with a text field for users who want to convey their wishes to bakers in writing. After completing these processes, the user clicks on the send request button at the bottom, and the request falls on the custom cake order page of the interested seller. The seller can make a price offer to the user by looking at the user's requests and information. The seller, who displays the cake information and user information requested by the user, enters the price he has determined in the price field at the bottom of the page. If it cannot meet the user's requests, there is also the option to refuse. After sending the price offer to the user, the custom cake is sent to the user's card. After the user sees the product on the user's card, they are directed to the address selection/entering screen before proceeding to the checkout step. In this section, the user has the chance to take advantage of the button where she/he can automatically select his address according to her/his location. Then, after coming to the ordering step, the user places an order after clicking the order button. After the order falls into the seller's system, it remains in the system until it is delivered. After the order is received by the user, the seller has successfully completed the order after confirming the order. This is how my custom cake system works.

4.5 System Screenshots:

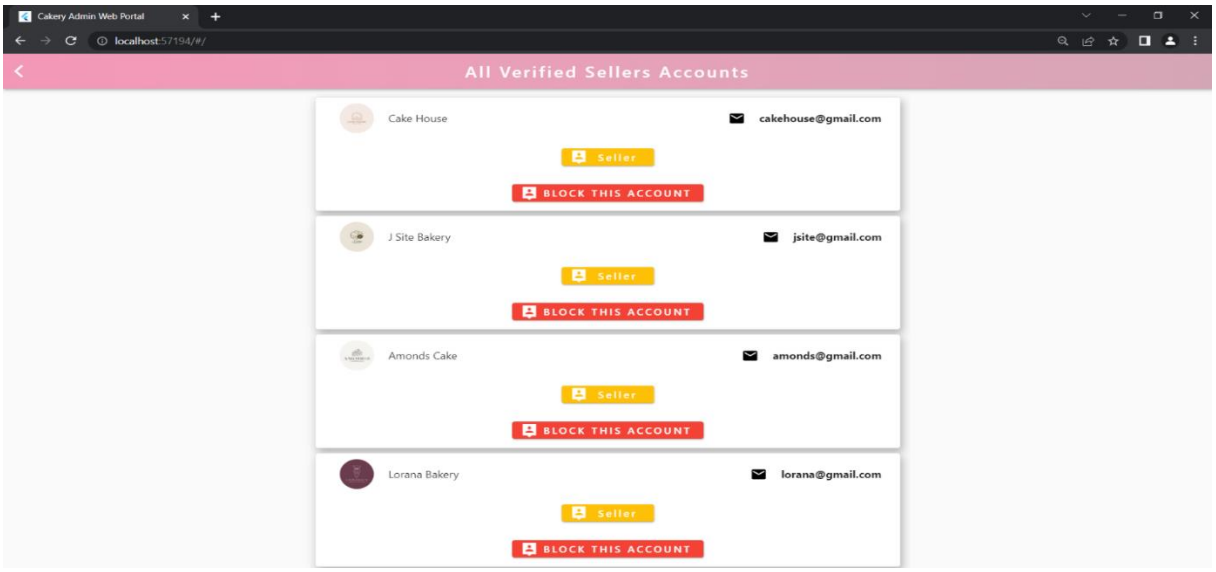
Admin Web Application



- **Admin Login Screen**

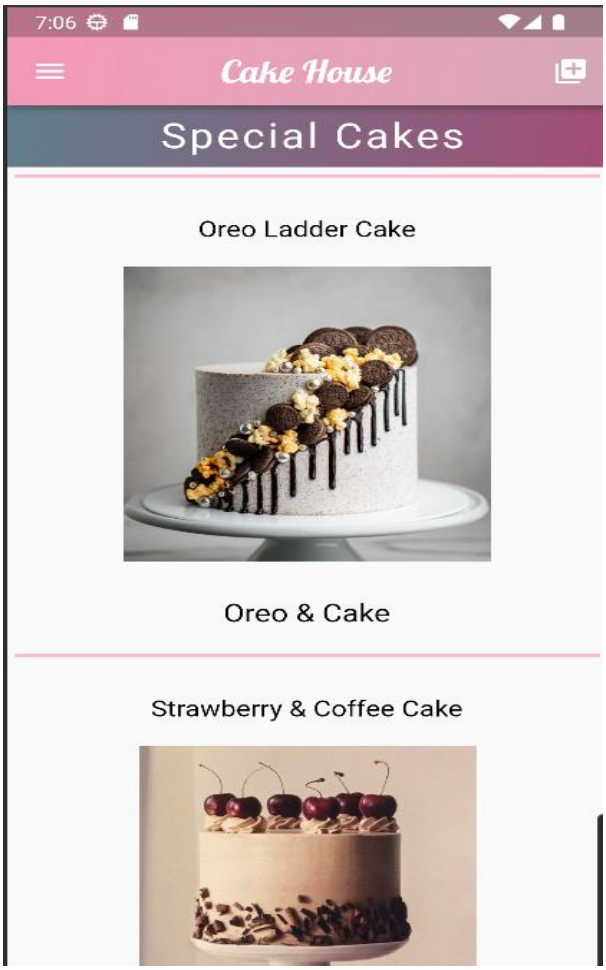


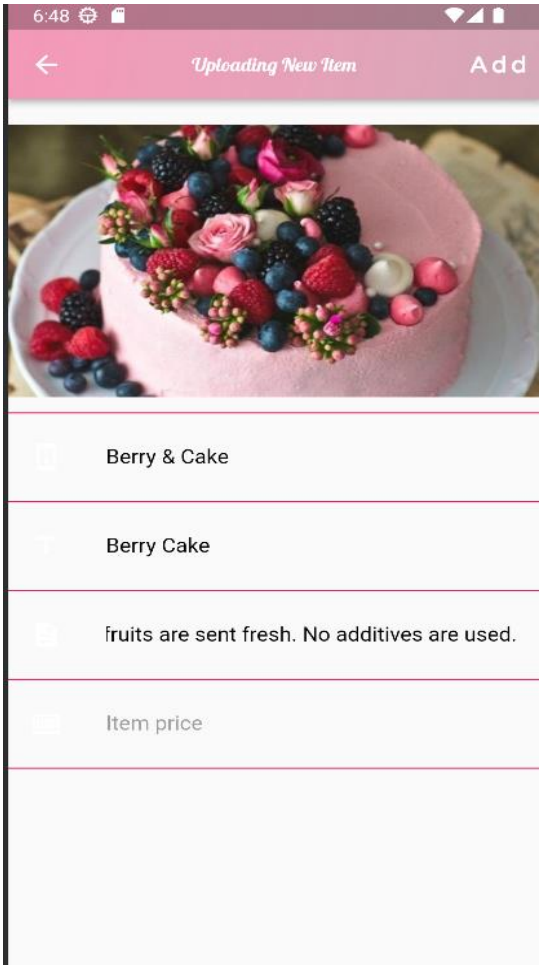
- **Admin Dashboard**



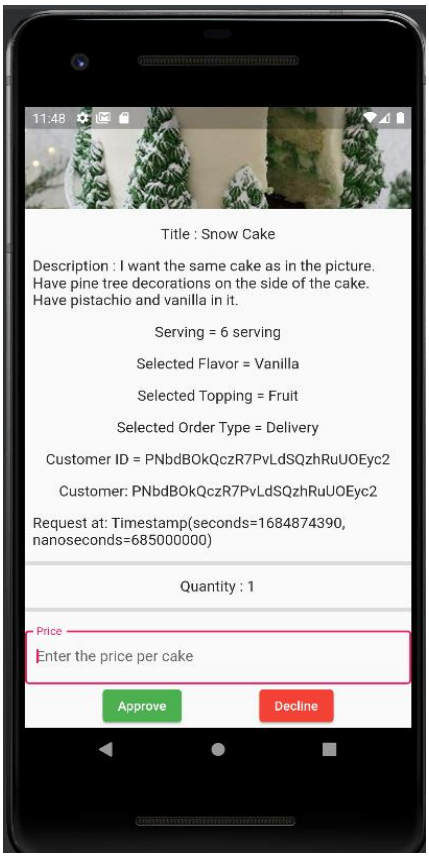
- The part where the admin can view the seller information and block the seller.

Cakery App (For Seller)





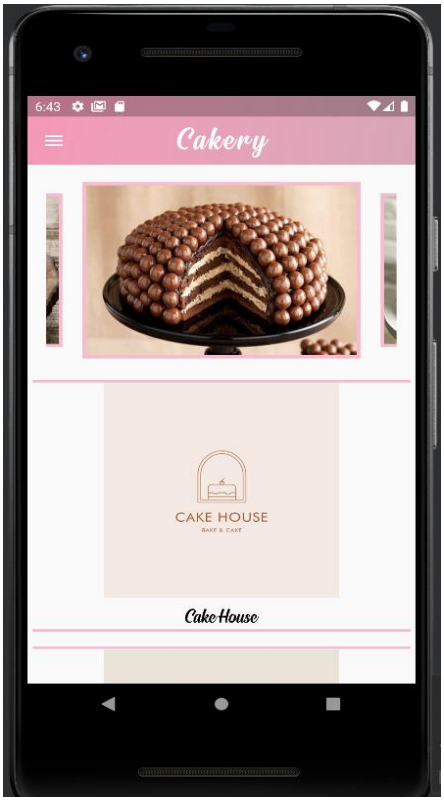
- Section where sellers can set their menu and items.



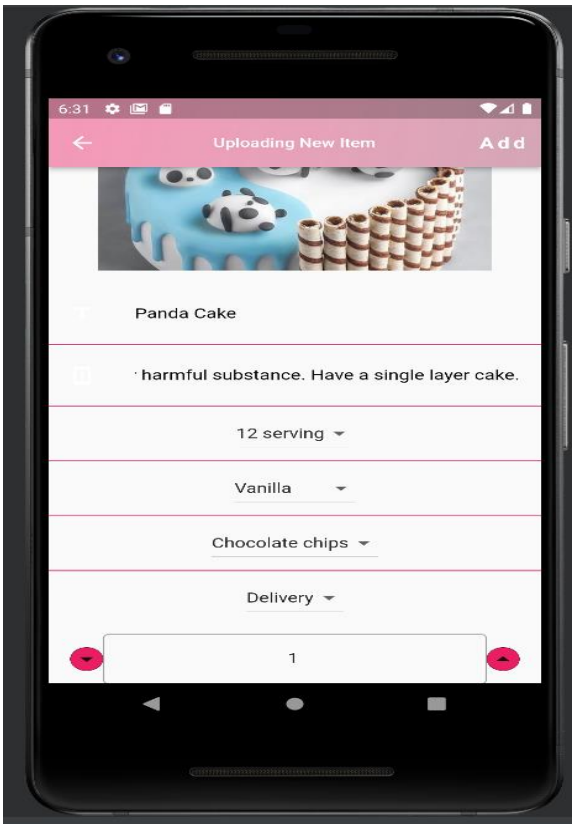
- The part where sellers view special cake requests from users.

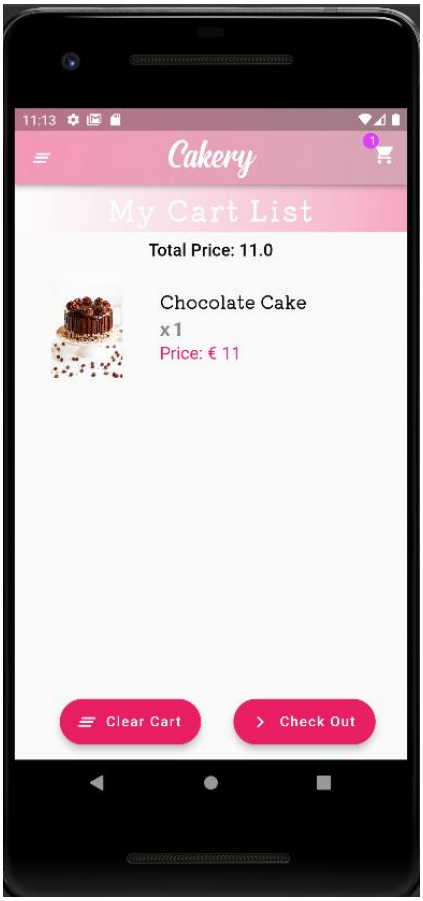
- Cakery App (For Users)

- The home page, which is the first page users see when they login the system.



- The page users use to order custom cakes a seller.





- Users cart page where user can view the items he/she wants to buy.

- The page where users see information about products and add them to the cart.

5. TEST RESULTS

We completed various tests to ensure our Cakery App performances in the desired way and meets all the requirements. As we mentioned in our test plan report our test cases cover modules including user registration and login, profile and dashboard display, ordering, and custom cake creation. All test cases were created to see the app's functionality and reliability while covering various scenarios.

User Display

Test Case ID	Test Scenario	Test Steps	Expected Results	Pass/Fail
UPD-001	User can view his/her profile	1. Login to the app 2. Click on the profile button	User profile page should be displayed with all the relevant details (name, email, phone number, address)	Pass
UPD-002	User can update his/her profile	1.Login to the app 2. Click on the "Profile" button 3. Click on the "Edit Profile" button 4. Update the relevant details 5. Click on the "Save" button	User profile details should be updated successfully	Pass
UPD-003	User can change his/her password	1.Login to the app 2. Click on "Profile" button 3. Click on the "Change Password" button 4. Enter the old password, new password and confirm password 5. Click on the "Save" button	User password should be changed successfully	Fail

Dashboard Display

Test Case ID	Test Scenario	Test Steps	Expected Results	Pass/Fail
DSD-001	User can view the dashboard	1. Login to the app 2. Click on the “Dashboard” button	User dashboard page should be displayed with all the relevant information (bakery cards, menu and item pictures)	Pass
DSD-002	User can view a bakery menu	1.Login to the app 2. Click on the “Dashboard” button 3. Click on one of the seller cards	User’s selected bakery menu information should be displayed with all the relevant details (menu image, title, description)	Pass
DSD-003	User can view a bakery item	1.Login to the app 2. Click on the “Dashboard” button 3. Click on one of the seller cards 4. Click on a menu	User’s selected bakery menu item information should be displayed with all the relevant details. (Item title, image, description, price)	Pass

Ordering

Test Case ID	Test Scenario	Test Steps	Expected Results	Pass/Fail
ORD-001	User can place an order	1. Login to the app 2. Select the desired bakery 3. Select the desired cake item 4. Add cake to the cart by clicking on the “Add to Cart” button 5. Proceed to checkout 6. Enter delivery details 7. Confirm the order	User should be able to place the order successfully and receive the confirmation message with the order ID	Pass
ORD-002	User can clear cart	1. Login to the app 2. Select the desired bakery 3. Select the desired cake item 4. Add cake to the cart by clicking on the “Add to Cart” button	User’s item should be cleaned from the cart successfully	Pass

		5. Click on the “Clear Cart” button		

Custom Cake Creation

Test Case ID	Test Scenario	Test Steps	Expected Results	Pass/Fail
CC-001	User can place a custom cake order	1. Login to the app 2. Select the desired bakery 3. Select the custom cake option 4. Fill the custom cake descriptions (num. of servings, flavor, topping choice) 5. Send it for approval	User should be able to place a custom cake order and send it to the bakery for approval	Pass
CC-002	Seller can approve/disapprove the custom cake order	1. Login to the seller app 2. Select the Custom cake item that a user send 3. Approve/Disapprove the item 4. Send it to user’s cart	Seller should be able to approve/disapprove the custom cake order coming from the user successfully	Pass

Add Location

Test Case ID	Test Scenario	Test Steps	Expected Results	Pass/Fail
ALC-001	User can add a new location	1. Login to the app 2. Click on the “Add Location” button 3. Enter the new location details (name, address, phone number) 4. Click on the “Save” button.	The new location should be added successfully and displayed in the user profile with the entered data	Pass
ALC-002	User can add location using Google Maps	1. Login to the app	User location should be added successfully and	Pass

		2. Click on the “Find My Location” button 3. Update the location details 4. Click on the “Save” button.	displayed in the user profile using the Google Maps	
ALC-003	User can delete an existing location	1. Login to the app 2. Click on the “My Addresses” button 3. Click on the “Delete” button 4. Click on the “Confirm” button.	The location should be deleted successfully and removed from the user profile	Fail

Menu and Item Creation

Test Case ID	Test Scenario	Test Steps	Expected Results	Pass/Fail
MC-001	Seller can add a new menu	1. Login to the seller app 2. Navigate to the menu management page 3. Click on the “Add Menu” button 4. Enter the menu name 5. Click on the “Save” button	Seller should be able to add a new menu category into their bakery	Pass
MC-002	Seller can add a new menu item	1. Login to the seller app 2. Navigate to the menu management page 3. Click on the “Add Item” button 4. Enter the item details (name, image, price, description)	Seller should be able to add a new item into their bakery menu with the entered details	Pass

		5. Click on the "Save" button		
MC-003	Seller can delete a menu item	1. Login to the seller app 2. Navigate to the menu management page 3. Click on the "Delete" button next to an existing item	Any item can be deleted from the bakery's menu and changes should be saved and reflected in the menu.	Pass

History of Orders

Test Case ID	Test Scenario	Test Steps	Expected Results	Pass/Fail
HO-001	Seller can view the history of orders	1. Login to the seller app 2. Navigate to the order history page	Seller should be able to view a list of all past orders with relevant details (order ID, customer details, order items, order status)	Pass

New Orders

Test Case ID	Test Scenario	Test Steps	Expected Results	Pass/Fail
NO-001	Seller can view the new orders	1. Login to the seller app 2. Navigate to the new order page	Seller should be able to view a list of all new orders that have not been proceed with relevant details (order ID, customer details, order items, order status)	Pass
NO-002	Seller can update the status of a new order	1. Login to the seller app 2. Navigate to the new order page 3. Select an order to process 4. Update the order status to "in progress" or "completed"	Seller should be able to update the status of a new order so that order is proceeded and visible in the "Order History"	Pass

For test results we can say majority of the test cases that conducted during the testing phase are produced successful outcome. However, a small number of tests ended in failure. Overall, our project has %88.9 passing rate for the test cases (18 test cases applied, 16 passed, 2 failed). During the testing process, several bugs are identified and documented as well. Each bug tracked down and fixed so that it won't affect user experience and functionality and ensuring a smooth error free user experience throughout the ordering process. We are aware of that we have potential enhancements based on the testing results. With the right iterations and enhancements in the future, Cakery app provides valuable insights. Some of the urgent iterations could be, improve the failed test cases, provide a safe and secure online payment system, improved error handling and performance optimization. Our test results revealed the overall pieces of our project that are reliable and successful. While some of the test cases ended in failure and some bugs occurred, we still ensure a high-quality user experience throughout or app for our users.

6. CONCLUSION

In conclusion, we have successfully put together 3 subsystems that creates Cakery. With Cakery, we believe we provide a user friendly, high-level cake ordering process while saving you to spend your days and hours to order cakes. We are hoping to enhance local bakery visibilities and give them a wide range of customer base.

7. GLOSSARY

Admin:The person who is in charge of the Cakery App.

Cart:A basket where the items are added into.

Customized Cake:A cake which can be customized as buyer's wants and needs.

Dashboard: Home page of Cakery App.

Dart:Dart is a programming language developed by Google. It's an object-oriented language thatcan be used for a variety of applications including web and mobile development.

Flutter:Flutter is an open-source mobile application development framework created by Google. It uses the Dart programming language and provides a rich set of pre-built widgets and tools to help developers create high performance, visually attractive mobile applications for both Android and IOS platforms.

Firestore: Firestore is a cloud based NoSQL document database service offered by Google. It provides real-time data synchronization and querying capabilities, making it a popular choice for building mobile and web applications that require fast, reliable data storage and retrieval.

Item: Cakes offered in bakeries.

UI: UI stands for “user interface,” and refers to the graphical layout and design of a software application or website. A good UI design helps users interact with the application more efficiently and effectively, by providing clear and intuitive navigation, visual cues, and feedback.

User: Person who orders from the Cakery App.

Product: Cakes offered in bakeries.

Profile: Detail of information about the user.

Seller: Bakeries that are enrolled in Cakery application.

8. REFERENCES

[1] Abhishek Singh, Adithya R, Vaishnav Kanade, Prof. Salma Pathan“ ONLINE FOOD ORDERING SYSTEM” International Research Journal of Engineering and Technology (IRJET) 2018.

[2] Sarika Rane1 , Varad Potdar , Harsh Trivedi, Yash Ughade, “Designing White Box Test Cases for Online Food Delivery System”, Computer Engineering, Shah and Anchor Kutchii Engineering College, Mumbai, India, 2021.

[3] “Object-Oriented Software Engineering, Using UML, Patterns, and Java,” 2nd Edition, by Bernd Bruegge and Allen H. Dutoit, Prentice-Hall, 2004, ISBN: 0-13-047110-0.

[4] D. Kotelenets, Test Plan Template.

[5] “Cakery High-Level Design Report”

[6] Flutter, Testing Flutter Apps, https://docs.flutter.dev/testing

[7] Patel, Mayurkumar, "Online Food Order System for Restaurants" (2015). Technical Library. Paper 219

[8] Varsha Chavan, Priya Jadhav, Snehal Korade, Priyanka Teli, ”Implementing Customizable Online Food Ordering System Using Web Based Application”, International Journal of Innovative Science, Engineering Technology(IJISET) 2015.

[9] Khairunnisa K., “The application of wireless food ordering system”, MASAUM Journal of Computing, Vol.1 Issue 2, Sept. 2009.

[10] Flutter, Integration Testing, https://docs.flutter.dev/testing/integration-tests

[11] Modus Create, Introduction to Flutter Widget Testing, https://moduscreate.com/blog/introduction-to-flutter-widget-testing/

[12] GeeksForGeeks, Types of Software Testing, https://www.geeksforgeeks.org/types-software-testing/

[13] Atlassian, Different Types of Software Testing, https://www.atlassian.com/continuous-delivery/software-testing/types-of-software-testing

[14] https://pub.dev/