



**TED ÜNİVERSİTESİ**

**CMPE 491**

**Senior Design Project**

**PROJECT ANALYSIS REPORT**

**Selin Akyurt - 17878041578**

**Melisa Uzulu - 3906408574**

**Recep Berkay Engin - 22271682594**

## **Table of Contents**

- 1. Introduction**
- 2. Current system (if any)**
- 3. Proposed system**
  - 3.1 Overview**
  - 3.2 Functional Requirements**
  - 3.3 Non-functional Requirements**
  - 3.4 Pseudo requirements**
  - 3.5 System models**
    - 3.5.1 Scenarios**
    - 3.5.2 Use case model**
    - 3.5.3 Object and class model**
    - 3.5.4 Dynamic models**
    - 3.5.5 User interface - navigational paths and screen mock-ups**
- 4. Glossary**
- 5. References**

## **1. Introduction**

Time is the most valuable thing in our world today and when we need a little help with saving time, web and mobile applications are here to serve us. Our application “Cakery” will help customers to order cakes for their special celebrations and will save time with allowing you to reach your local bakeries with just one app. Our application will not only serve ready to eat cakes but will give you an opportunity to create your customized cake from the bakery you choose. Your cake order will arrive on the date and time you want. All you need to do is just place your order using Cakery App. The whole purpose of the application is solving the problem of going to bakeries, choosing a cake or ordering a customized cake, which takes more time and effort to order, and bring it home. The application will be available for mobile platforms and the project includes 3 different systems. We’re aiming to create an application that will consist of 3 systems. One system that users can go through bakery items in order to order a cake, one system that bakeries can control their selling’s and menu items and another system that an admin can activate or block accounts.

## **2. Current System**

At this stage we don’t have any current system.

## **3. Proposed System**

### **3.1 Overview**

The application aims to provide a service to the users where they will be able to order cakes and create customized cakes without having to go to bakeries. In another perspective, it will also help bakeries to reach out more customers and increase their overall sales. For this project we decided to have 3 different systems for different purposes of the application.

For the user part of the application, customers need to login to the system in order to place an order. If the user hasn’t created a profile, they can sign in using their email, password and telephone number. After creating their account, they can login to the application. Listing all the bakeries that registered in the system and customer choosing one of them, they will be able to see the items and prices based on the chosen bakery. They can choose an item, add it to their cart and continue their shopping if its needed. If the chosen bakery provides customized special day orders, they can place a cake order with the specific wants and needs, explaining what kind of a cake they want. The customers need to be specific about what kind of cake they want and give detailed information in order not to be disappointed with the outcome. Our application will help users to place a customized order with asking to answer specific question at this stage. They can also describe the cake using a text box or place a reference photo.

After the customers deciding what they want to order, they can go to their cart and see the total price and finish the ordering process or they can clear the cart. For the final check out process, customers have to give their address information. They can write the address on their

own or using their current location as an address. They can save the address information for later use. Users can have more than one address information saved on their profile, and add a new address from their profile. After selecting the address, user will be directed to a payment page in order to complete the order. After placing an order, users can see their current orders from my orders page or they can see all their past orders from the history page. User can logout from their account whenever they want.

For the seller part of the application, bakeries have to register to the system. After that they can put their items on their page, delete them later or add a new item whenever they want. Bakeries must give detailed information about the cake as well as the price. If the bakery serves customized cakes, they can create a customized cake item where users can create their own cake. Seller app will include home page, earnings page where they can see the total earning of the bakery, new orders where they can see the orders they haven't delivered yet and history page where they can see the all the past orders that has been delivered. In the new orders page, sellers will be able to see the delivery address, delivery date and time, and payment information with the order id. When the order is delivered, the seller has to confirm that order is done. When an order has been ticked as done, it will be no longer seen in the new order page and will be moved to order history. Sellers can logout from their account whenever they want.

For the admin part, we decided that it will be more useful as a web page rather than an application. So, the admin web portal will be able to see all the user and seller information, activate user/seller accounts and block user/seller accounts in case of any misuse. Admin can also see the total earnings of each bakery.

### 3.2 Functional Requirements

Number	Functional Requirement Explanation	Admin Web Portal (A)
Req-1	Admin will enter their e-mail.	<i>1. Login Page Module</i>
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-6	Admin will click on the "Register" button if they don't have an account.	
Req-7	If the e-mail is already taken the admin gets an error message.	<i>2. Register Page Module</i>
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	Admin fills in the confirm password information in the required field.	
Req-13	If admin enters an incorrect e-mail address, admin will receive an error message.	
Req-14	Admin can upload photos to required field.	
Req-15	When all the necessary information is filled, the information is saved and transferred to the main page by clicking on the save part	
Req-16	If there is any missing information, it will encounter an error message.	
Req-17	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-18	If the newly entered password meets the required password qualifications, the confirmation will take place	
Req-19	If it does not meet the required password qualifications, it will get an error message.	
Req-20	Admin can view active user accounts.	<i>3. Admin Portal Page Module</i>
Req-21	Admin can view active seller accounts.	
Req-22	Admin can view block user accounts.	
Req-23	Admin can view block seller accounts.	
Req-24	Admin can see the date, time, and month information on the screen.	
Req-25	If admin wants to exit the portal page, admin clicks the logout button.	
Req-26	Admin can click to view active user accounts.	<i>3.1 Active User Accounts Page Module</i>
Req-27	Admin can view the user's name-surname.	
Req-28	Admin can see the user's phone number.	
Req-29	Admin can view user's photo.	
Req-30	Admin can click "Block this account" button to remove the user from the system.	
Req-31	If the admin blocks the user, the user is displayed on the blocked user's page.	
Req-32	Admin can click to view block user accounts.	<i>3.2 Block User Accounts Page Module</i>
Req-33	Admin can view the block user's name-surname.	
Req-34	Admin can see the block user's phone number.	
Req-35	Admin can view user's photo.	
Req-36	If the admin wants to add the user back to the system, they can click the "Activate this account" button.	
Req-37	The user that the admin reactivated is displayed in the "active user accounts" section.	
Req-38	Admin can click to view active seller accounts.	

Req-39	Admin can view the company names of the active sellers.	<i>3.3 Active Seller Accounts Page Module</i>
Req-40	Admin can see the active seller's phone number.	
Req-41	Admin can view the seller's photo.	
Req-42	Admin can click "Block this seller" button to remove the seller from the system.	
Req-43	Admin can view the total earnings of the seller.	
Req-44	If the admin blocks the seller, the user is displayed on the blocked seller's page.	
Req-45	Admin can click to view block seller accounts.	<i>3.4 Block Seller Accounts Page Module</i>
Req-46	Admin can view the company names of the block sellers.	
Req-47	Admin can see the block seller's phone number.	
Req-48	Admin can view block seller's photo.	
Req-49	If the admin wants to add the seller back to the system, they can click the "Activate this seller" button.	
Req-50	The seller that the admin reactivated is displayed in the "active seller accounts" section.	

Number	Functional Requirement Explanation	Cakery App (B)
Req-51	User will enter their e-mail.	<i>4. User Login Page Module</i>
Req-52	User will enter their password.	
Req-53	User interacts with login button to log in the system.	
Req-54	If the e-mail is invalid, user gets error message.	
Req-55	If the password is invalid, user gets error message	
Req-56	If the user forgets the password, click the "Forgot password" button for help.	
Req-57	User will click on the "Create an Account" button if they don't have an account.	
Req-58	If the e-mail is already taken the user gets an error message.	<i>5. User Register Page Module</i>
Req-59	If the user leaves the password blank, an error message will appear.	
Req-60	User fills in the name and surname information in the required field.	
Req-61	User fills in the email information in the required field.	
Req-62	User fills in the password information in the required field.	
Req-63	User fills in the confirm password information in the required field.	
Req-64	If user enters an incorrect e-mail address, user will receive an error message.	

Req-65	User can upload photos to required field.	
Req-66	When all the necessary information is filled, the information is saved and transferred to the main page by clicking on the save part.	
Req-67	If there is any missing information, it will encounter an error message.	
Req-68	User can see the current profile information in the input boxes	<i>6. User Profile</i>
Req-69	User can see the name-surname.	
Req-70	User can see the e-mail.	
Req-71	User can see the telephone number.	
Req-72	User can see the country.	
Req-73	User can see the city.	
Req-74	User can see own address.	
Req-75	User can see the current profile photo.	
Req-76	User can change their own address.	<i>6.1. Edit User Profile</i>
Req-77	The user can see the "information" section where user can view her/his own information.	<i>7. User Dashboard Module</i>
Req-78	If there is a discount, the user can view the discount information.	
Req-79	User can view featured sellers.	
Req-80	User can view all sellers.	
Req-81	Users can view menu when they click on the sellers.	
Req-82	Users can see their name and surname.	<i>7.1 User Information Module</i>
Req-83	Users can see their name and surname.	
Req-84	Users can view the current order with the "my orders" button.	
Req-85	Users can view the past order with the "history" button.	
Req-86	Users can search for vendors or products with the search button.	
Req-87	Users can "add new address" new addresses.	
Req-88	Users can exit the system with the logout button.	
Req-89	User fills in the name and surname part appropriately.	<i>7.1.1 Add New Address Module</i>
Req-90	User fills in the phone number section appropriately.	
Req-91	User fills in the address part appropriately.	
Req-92	User fills in the city part accordingly.	
Req-93	User fills in the country part appropriately.	
Req-94	User fills in the full address part accordingly.	

Req-95	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-96	The user can see the menu of a seller she/he wants.	<i>7.2 Menu View Module</i>
Req-97	Users can view all products in the menu.	
Req-98	Users can see the prices of the products.	
Req-99	Users can view information about products.	
Req-100	Users can select the number of products to be added to the cart.	
Req-101	Users can click the "add to cart" button to add the product to the cart.	
Req-102	Users can see the photo of the product.	
Req-103	Users can see how many items are in the cart.	
Req-104	Users can view the name of the product.	
Req-105	The user can view the rating of the product.	
Req-106	If the user wants to make a customized product, he can click the customized cake option.	<i>7.2.1 Customize Menu Module</i>
Req-107	The customer who clicks on the customized cake option goes to the customized product's customization page.	
Req-108	The user can see/upload the picture of the product on this page.	
Req-109	The user can see the name of the product.	
Req-110	The user can see the size of the product and can choose between sizes.	
Req-111	The user can choose from among the materials to be added to the product as he wishes.	
Req-112	The user can see the price of the product.	
Req-113	When the user adds more products, they can see the updated price of the product.	
Req-114	The user can add the product he/she has prepared to the basket with the "Add To Cart" button.	
Req-115	The user can press the back button to view another product.	
Req-116	The user can press the delete button to delete the extra materials selected for the product.	<i>7.3 Cart Page Module</i>
Req-117	Users will be able to view all the products added to the cart here.	
Req-118	Users will be able to see the name of the product.	
Req-119	Users will be able to see the quantity of the product.	
Req-120	Users will be able to see the price of the product.	
Req-121	If users want to add more products, there will be an add button.	
Req-122	Users will be able to click on the "Clear Cart" button if they want to remove all the products in the cart.	
Req-123	Users will be able to see the total cart amount.	



Req-124	Users can use the checkout button to go to the payment page.	
Req-125	Users will be able to see the products they will order.	<i>7.3.1 Payment Page Module</i>
Req-126	Users will be able to choose their registered home address.	
Req-127	Users will be able to add new addresses if they do not have registered home addresses.	
Req-128	Users will be able to use their current location addresses instead of their registered home addresses.	
Req-129	Users will be able to choose between different payment methods, online or in cash.	
Req-130	Users can see the shipping cost.	
Req-131	Card information will be requested from users who choose to pay online.	
Req-132	Users will be able to fill in the card information section appropriately.	
Req-133	Users will have successfully placed an order after making the payment.	
Req-134	Users will be able to view information about successful order.	<i>7.3.2 Successful Order Page Module</i>
Req-135	Users will be able to view the amount of the order.	
Req-136	Users will be able to view the stage of the order.	
Req-137	Users will be able to see the address to which the order will be sent.	
Req-138	Users will be able to see the name and surname of the person who placed the order.	
Req-139	Users will be able to see the order date and time.	
Req-140	Users will be able to see the order ID.	
Req-141	Users will be able to see the phone number.	
Req-142	Bakery will enter their e-mail.	<i>8- Bakery Login Module</i>
Req-143	Bakery will enter their password.	
Req-144	Bakery interacts with login button to log in the system.	
Req-145	If the e-mail is invalid, bakery gets error message.	
Req-146	If the password is invalid, bakery gets error message.	
Req-147	If the bakery forgets the password, click the "Forgot password" button for help.	
Req-148	Bakery will click on the "Create an Account" button if they don't have an account.	
Req-149	Bakery will enter their e-mail.	
Req-150	If the e-mail is already taken the bakery gets an error message.	
Req-151	If the bakery leaves the password blank, an error message will appear.	
Req-152	Bakery fills in the name information in the required field.	
Req-153	Bakery fills in the email information in the required field.	

Req-154	Bakery fills in the password information in the required field.	<i>9- Bakery Register Module</i>
Req-155	Bakery fills in the confirm password information in the required field.	
Req-156	If bakery enters an incorrect e-mail address, bakery will receive an error message.	
Req-157	Bakery can upload photos to required field.	
Req-158	When all the necessary information is filled, the information is saved and transferred to the main page by clicking on the save part.	
Req-159	If there is any missing information, it will encounter an error message.	
Req-160	When the bakery wants to change the password, bakery enters the password in the password field and confirms it in the confirm old password to save field.	
Req-161	If the newly entered password meets the required password qualifications, the confirmation will take place.	
Req-162	If it does not meet the required password qualifications, it will get an error message.	
Req-163	The bakery can see the "information" section where bakery can view her/his own information.	<i>10-Bakery Dashboard Module</i>
Req-164	Bakery can see its items/menu on listed order.	
Req-165	Bakery can view details when it clicks on an item.	
Req-166	Bakery can see a “Add New Item” button on right top.	
Req-167	Bakery can see items in descending order.	
Req-168	Bakery can see their name.	<i>11-Bakery Information Module</i>
Req-169	Bakery can see their logo picture.	
Req-170	Bakery can view its total earnings with the “My Earnings” button.	
Req-171	Bakery can view the past order with the "History" button.	
Req-172	Bakery can view the present orders with the "New Orders" button.	
Req-173	Bakery can see its home page with the “Home” button.	
Req-174	Bakery can exit the system with the logout button.	
Req-175	If there’s a new order visible, bakery can click on the order to see detailed information of the order.	<i>11.1-Bakery Orders Module</i>
Req-176	Bakery can see shipping details of new order.	
Req-177	Bakery can see the price of new order.	
Req-178	Bakery can see history of orders from “History”.	
Req-179	Bakery can see amount of an item that has recently sold.	
Req-180	Bakery can see price of an item that has recently sold.	

Req-181	Bakery can see it's total earnings from "My Earnings."	<i>11.2 - Bakery Earnings Module</i>
Req-182	Bakery can create/adjust its Menu.	<i>12 - Bakery Edit Menu Module</i>
Req-183	Bakery can create an item for it's menu by clicking on the "Add New Item" button on top right.	<i>12.1- Bakery Create Item Module</i>
Req-184	Bakery can upload a picture of the new item.	
Req-185	Bakery can fill in item information.	
Req-186	Bakery can set a price for the new item.	
Req-187	Bakery can delete an item by selecting delete icon that is displayed in detailed screen.	<i>12.2- Bakery Edit Menu Module</i>

### 3.3 Non-functional Requirements

Number	Non-Functional Requirement Explanation	Cakery App
Req-1	System should support 1000 customers at a time.	
Req-2	The system must comply with Microsoft / Apple / Google accessibility guidelines.	
Req-3	Screens should respond quickly to users.	
Req-4	A crash in system shouldn't result in loss of data.	
Req-5	Application should be available for use 7/24.	
Req-6	System should provide data protection and security.	
Req-7	Only admins have the authority to update records and sensitive data.	
Req-8	The system should run stably for every user without crashing.	
Req-9	System should be able to receive information, save in database and interact with requests periodically.	
Req-10	Systems should be interdependent of each other.	
Req-11	Application should be easy to learn and use in all aspects.	

### 3.4 Pseudo Requirements

No	Pseudo Requirement Explanation
<i>Req-1</i>	The system should be implemented with Flutter.
<i>Req-2</i>	The location information should be drawn from Google Maps
<i>Req-3</i>	The system database should be implemented with NoSQL.

### **3.5 System Models**

#### **3.5.1 Scenarios**

##### **3.5.1.1. Cake Ordering System**

User orders a cake from Cakery Application.

- User login to the application.
- Different bakeries are displayed in the main page.
- User choses a bakery.
- The uploaded items from the chosen bakery are displayed on the screen.
- User clicks on the item from the menu displayed.
- User sees the price and the description of the item clicked.
- User adds the item to cart.
- User adds the item that can be customized.
- User fills out the customization form.
- User adds the item to cart.
- If the cart contains correct and wanted items, user approves the cart.
- User is directed to address selection page.
- User gets current location information from Google Maps.
- User selects payment method as cash on delivery.
- User is directed to order placed successful page and sees shipping details.

##### **3.5.1.2. Bakery Edit Menu and View Order System**

Bakery edits the menu and views current order requests.

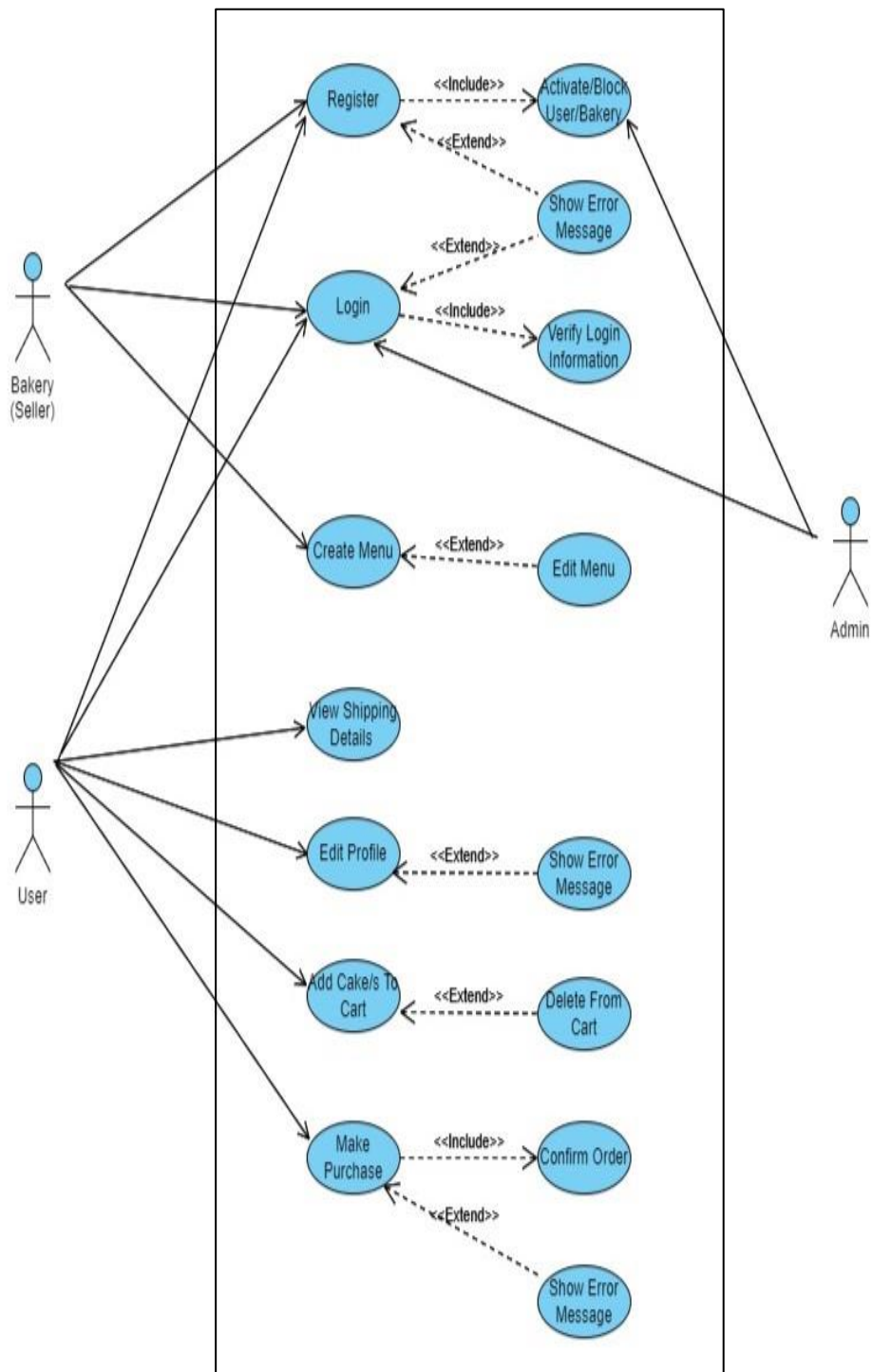
- Bakery login to the application.
- Bakery choses add new item.
- Bakery adds picture, name, price and description information.
- Bakery checks new orders.
- Bakery ships the package when order is ready.
- Bakery confirms the order.

##### **3.5.1.3. Admin Activate/ Block System**

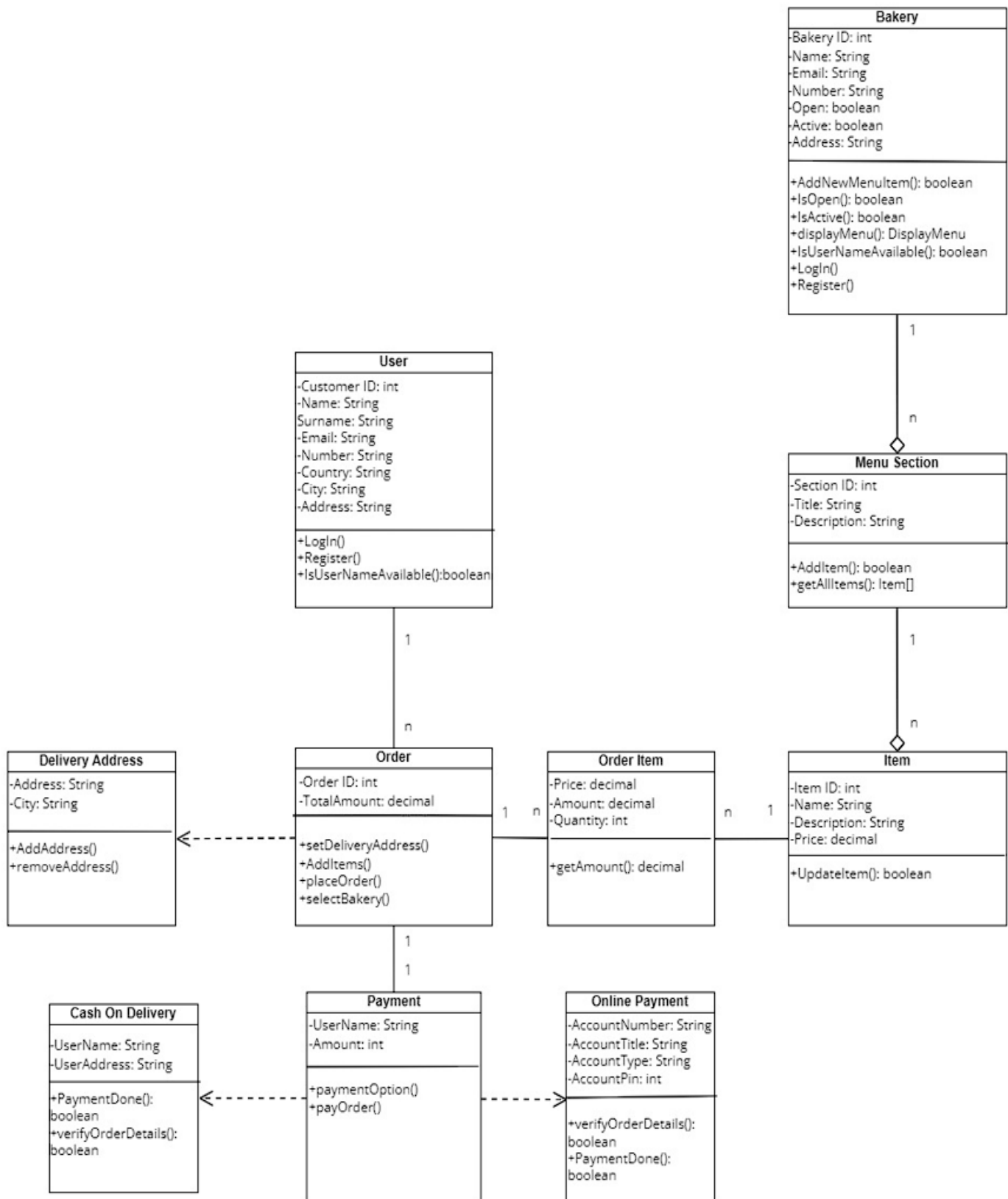
Admin activates or blocks user/seller accounts.

- Admin login to portal.
- Admin selects activate user accounts.
- Admin choses an account.
- Admin activates user account.
- Admin selects block seller accounts.
- Admin choses a bakery.
- Admin blocks the bakery due to Cakery App Bakery Policy.

### 3.5.2 Use-Case Model



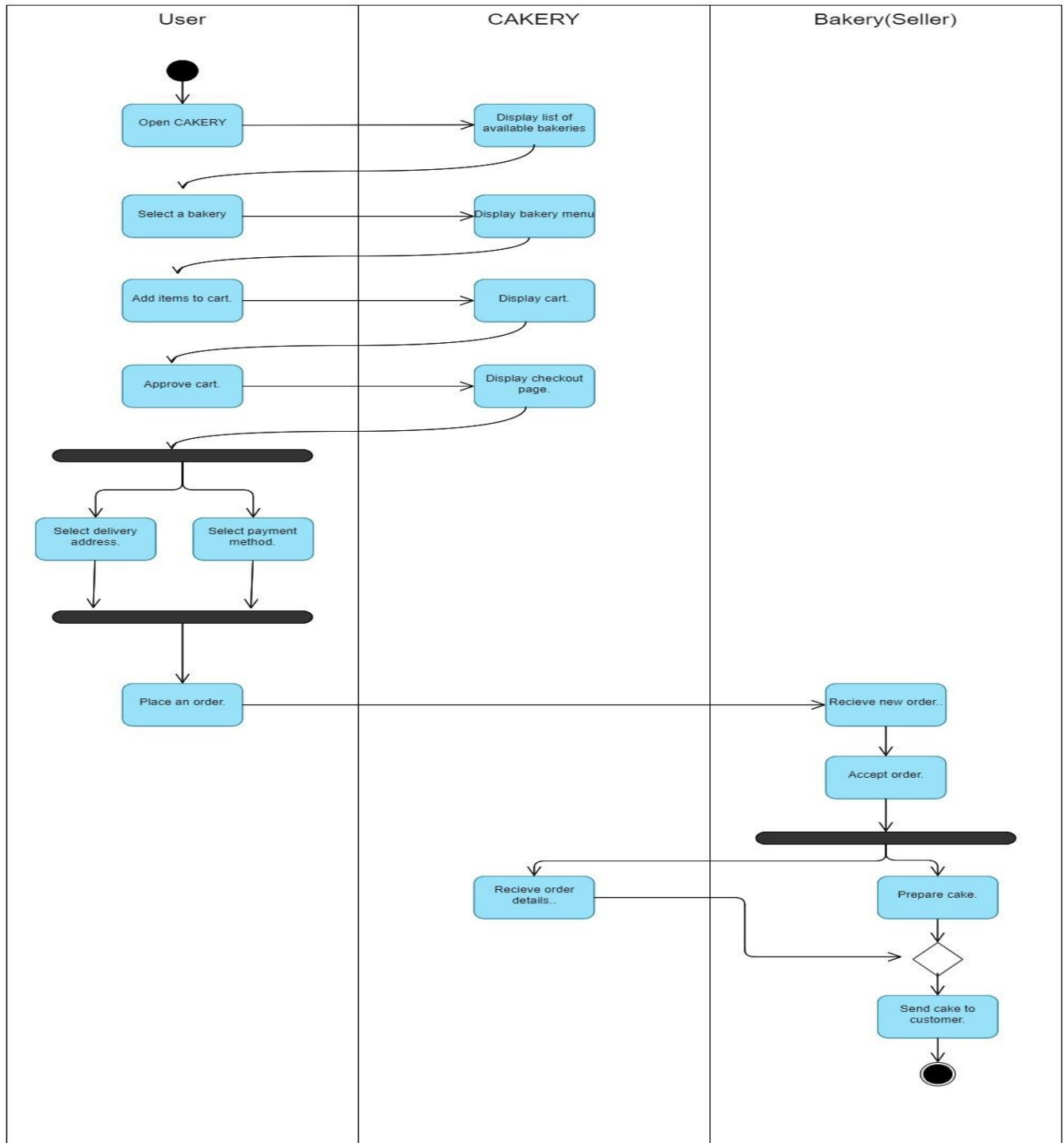
### 3.5.3 Object and Class Model



### 3.5.4 Dynamic Models

<b>A</b>	<b>Admin Web Portal</b>
<i>1</i>	<i>Login Page Module</i>
<i>2</i>	<i>Register Page Module</i>
<i>3</i>	<i>Admin Portal Page Module</i>
<i>3.1</i>	<i>Active User Accounts Page Module</i>
<i>3.2</i>	<i>Block User Accounts Page Module</i>
<i>3.3</i>	<i>Active Seller Accounts Page Module</i>
<i>3.4</i>	<i>Block Seller Accounts Page Module</i>
<b>B</b>	<b>Cakery App</b>
<i>4</i>	<i>User Login Page Module</i>
<i>5</i>	<i>User Register Page Module</i>
<i>6</i>	<i>User Profile</i>
<i>6.1</i>	<i>Edit User Profile</i>
<i>7</i>	<i>User Dashboard Module</i>
<i>7.1</i>	<i>User Information Module</i>
<i>7.1.1</i>	<i>Add New Address Module</i>
<i>7.2</i>	<i>Menu View Module</i>
<i>7.2.1</i>	<i>Customize Menu Module</i>
<i>7.3</i>	<i>Cart Page Module</i>
<i>7.3.1</i>	<i>Payment Page Module</i>
<i>7.3.2</i>	<i>Successful Order Page Module</i>
<i>8</i>	<i>Bakery Login Module</i>
<i>9</i>	<i>Bakery Register Module</i>
<i>10</i>	<i>Bakery Dashboard Module</i>
<i>11</i>	<i>Bakery Information Module</i>
<i>11.1</i>	<i>Bakery Orders Module</i>
<i>11.2</i>	<i>Bakery Earnings Module</i>
<i>12</i>	<i>Bakery Edit Menu Module</i>
<i>12.1</i>	<i>Bakery Create Item Module</i>
<i>12.2</i>	<i>Bakery Edit Menu Module</i>

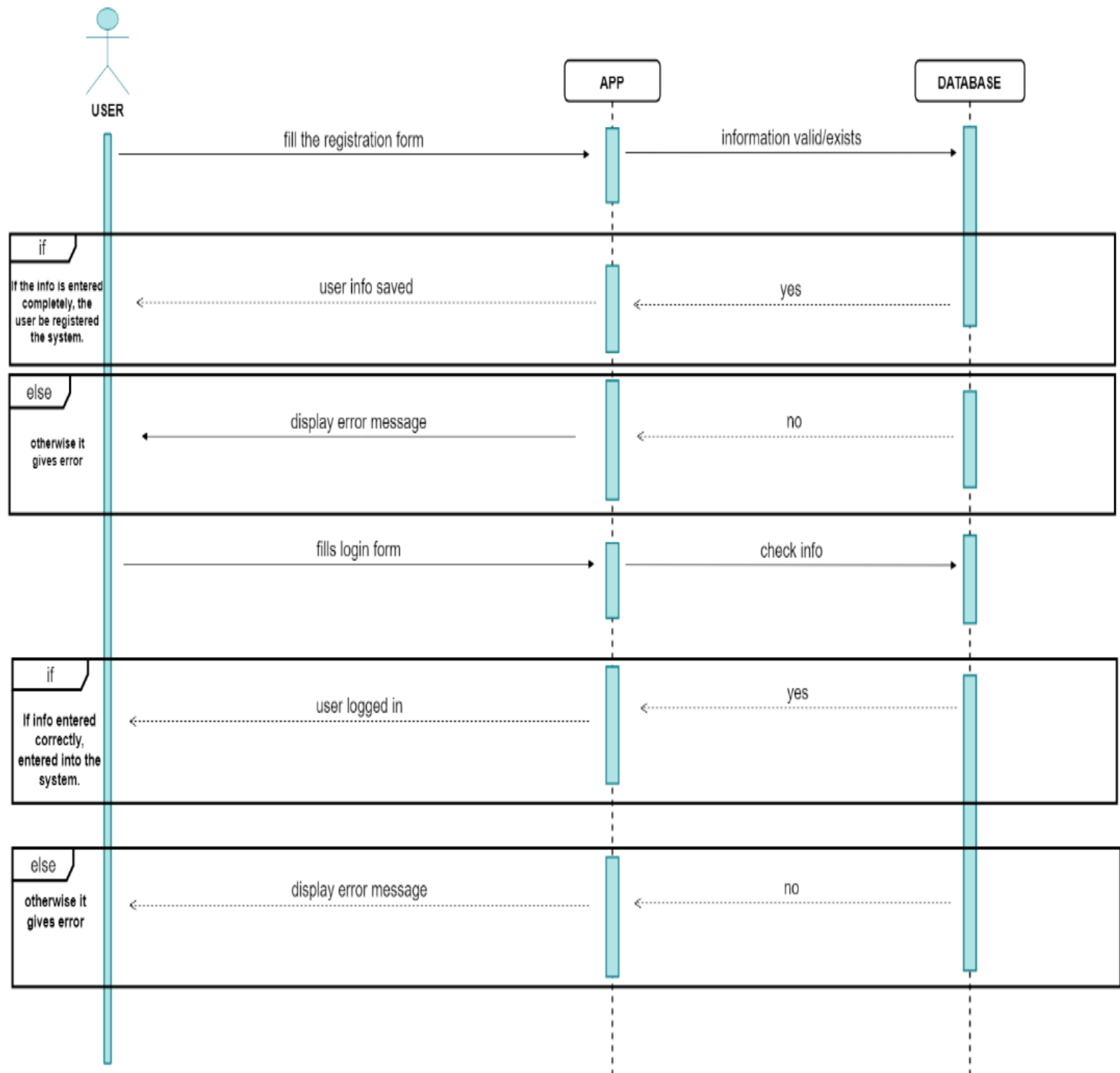
## State Diagram



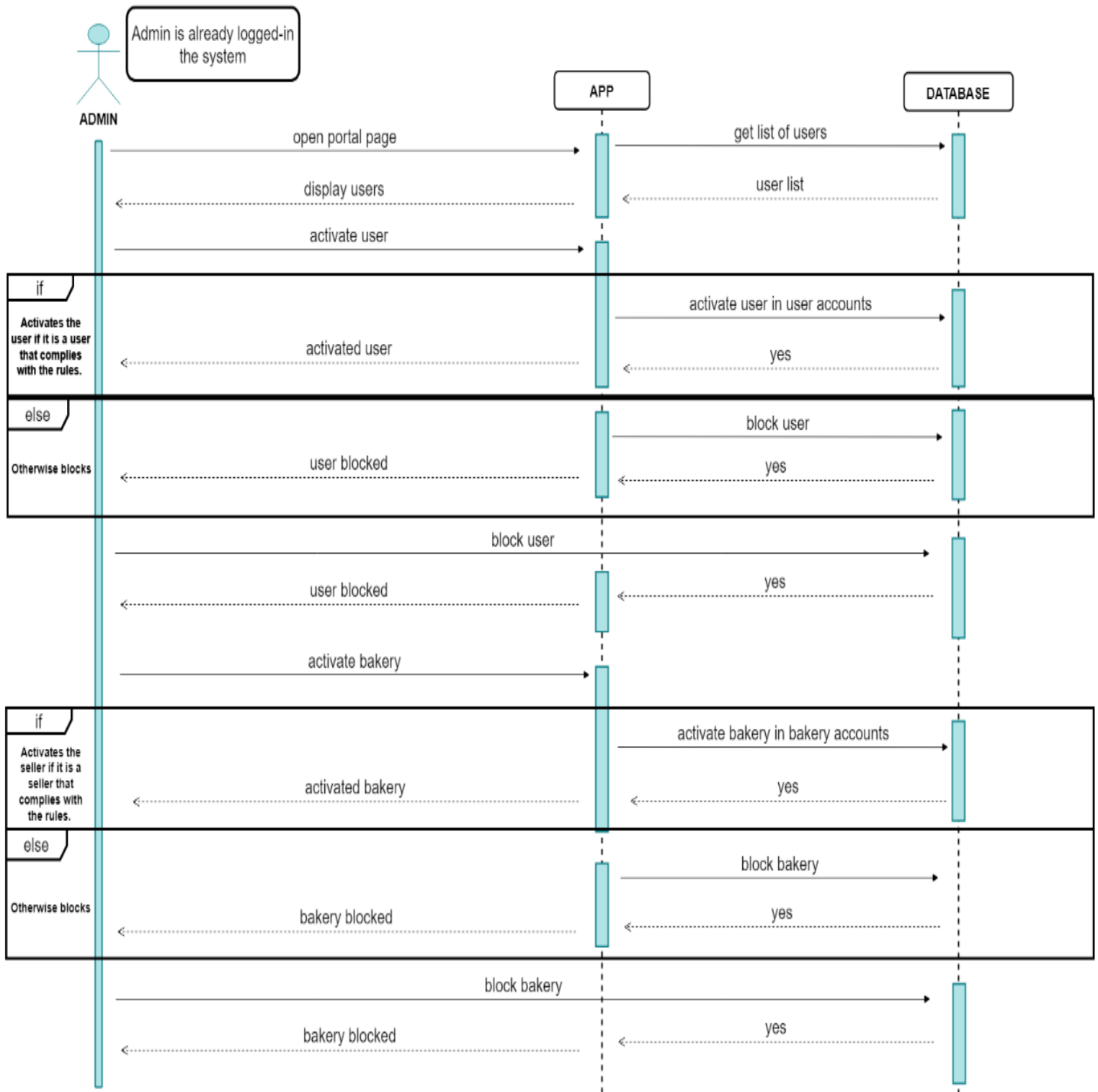
## Sequence Diagrams



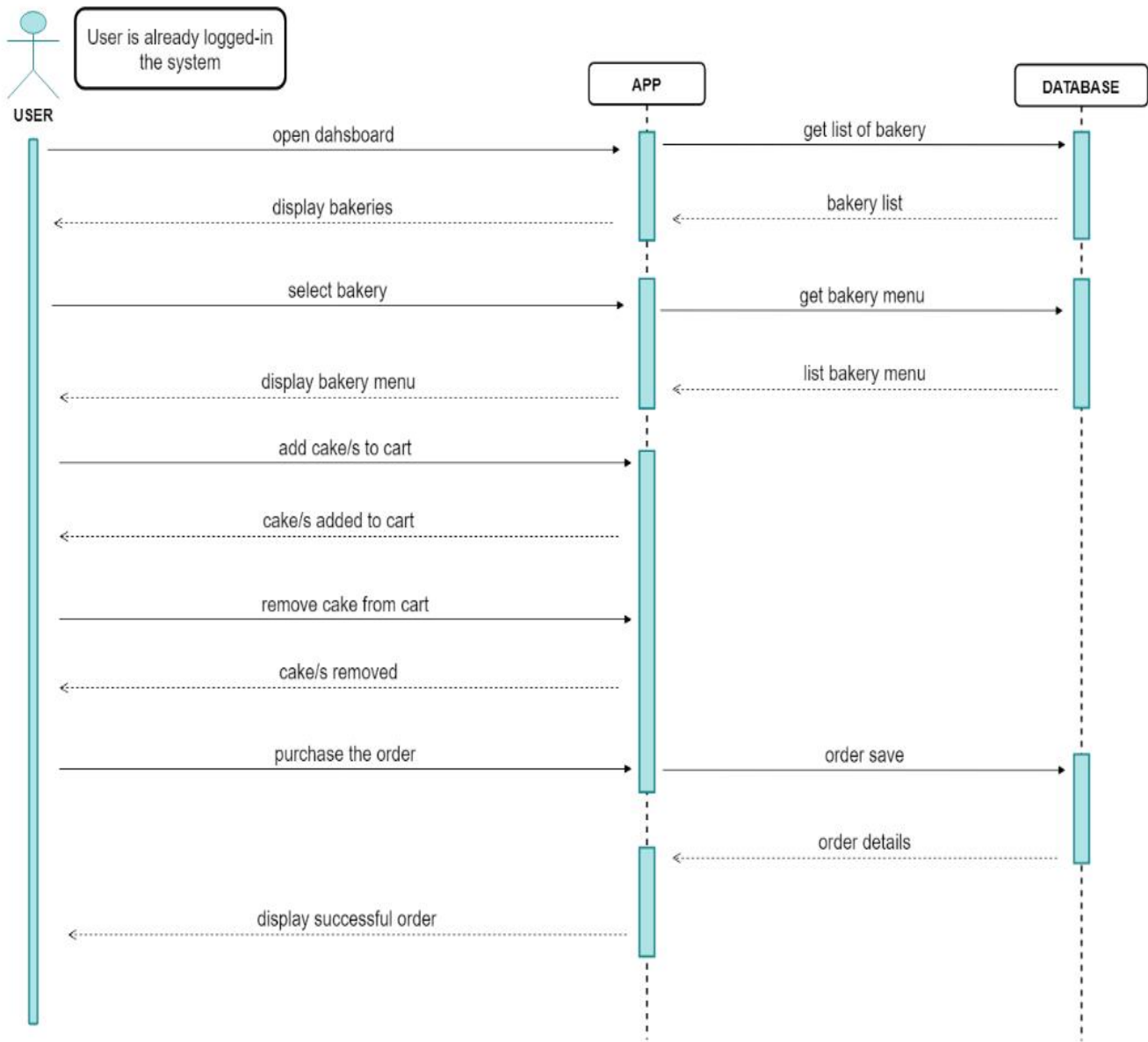
## 1. Login Module



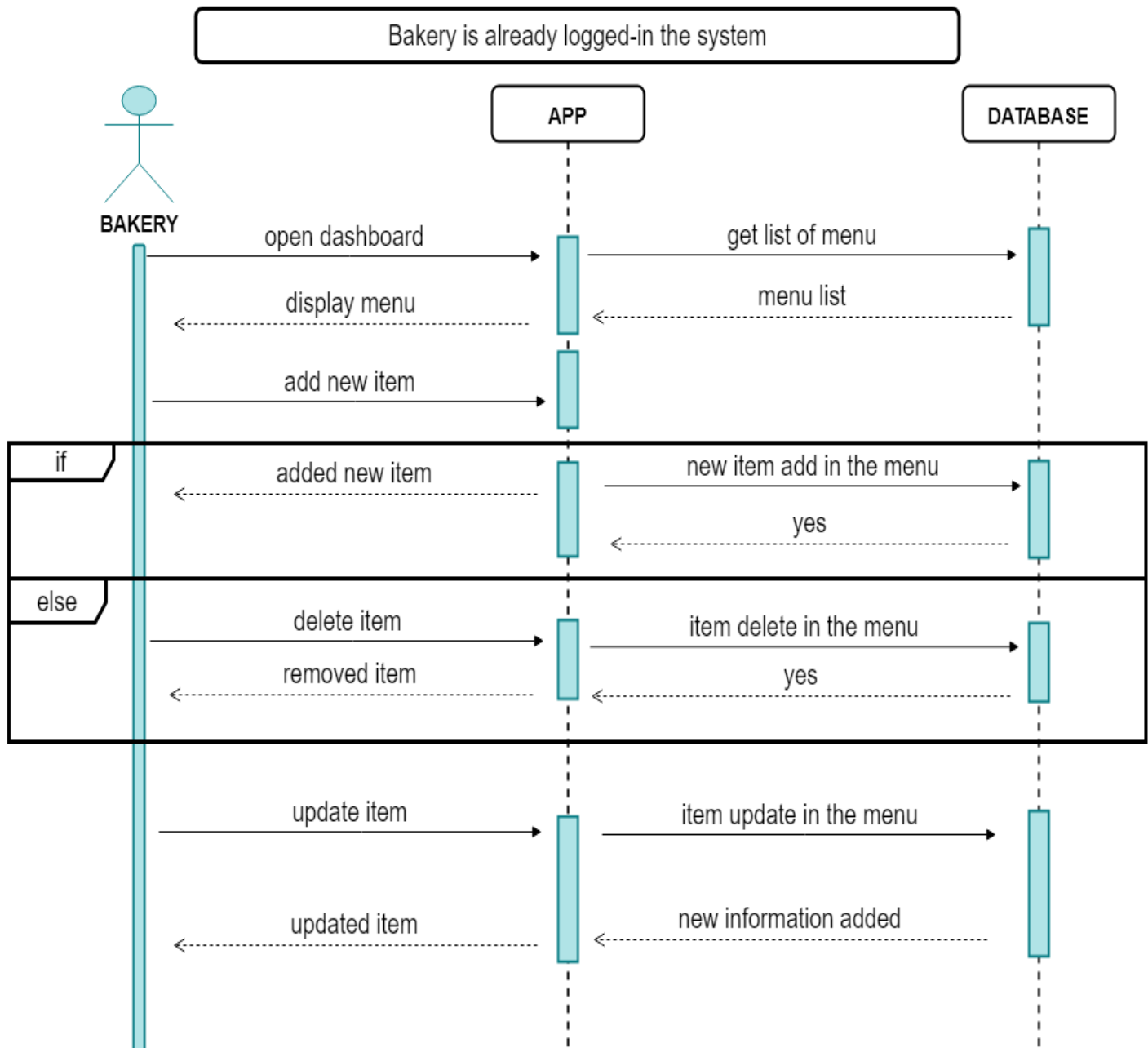
## 2. Admin Portal Module



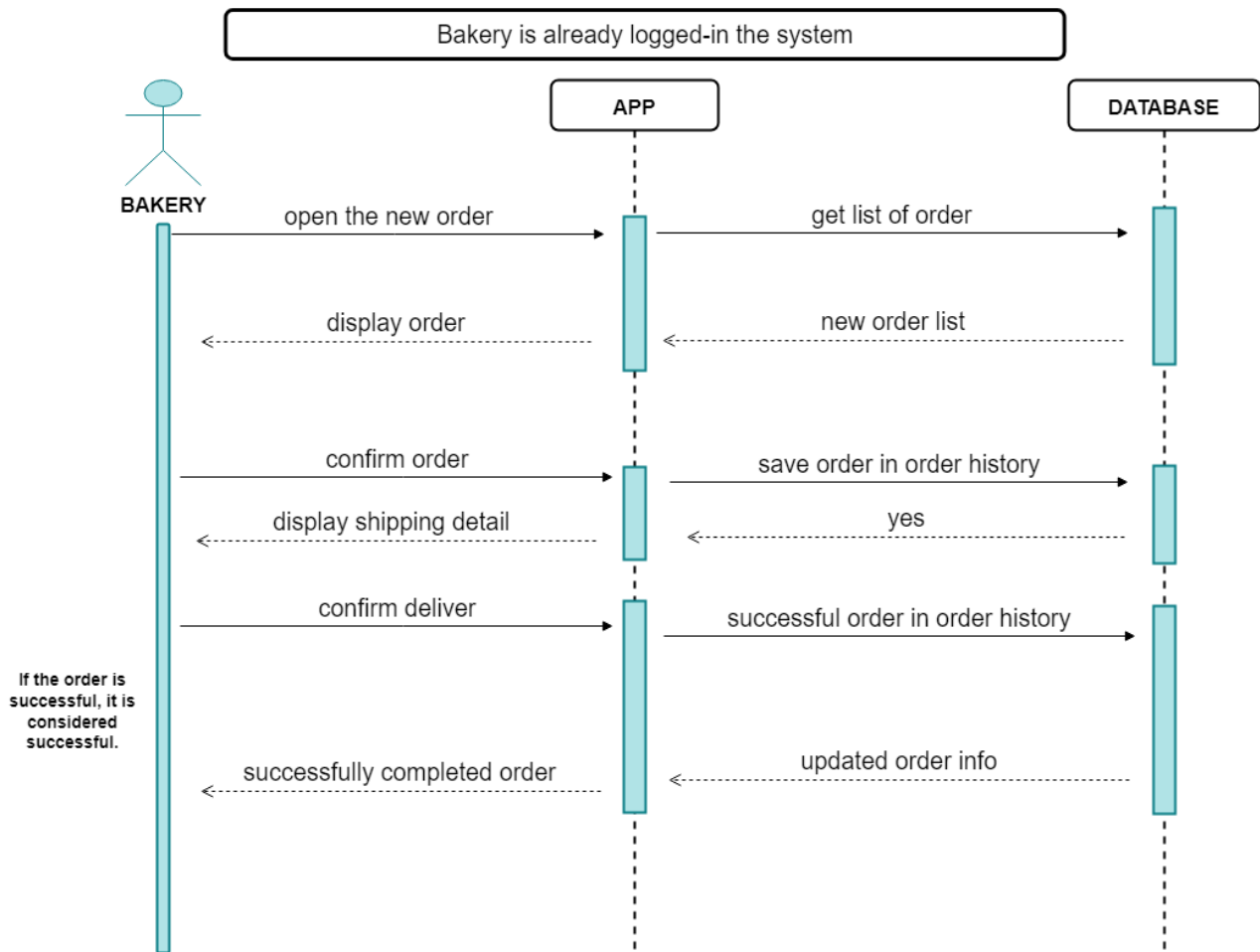
### 3. User Shopping Module



#### 4. Bakery Add Item Module



## 5. Order Module



### 3.5.5 User Interface - Navigational Path and Screen Mock Ups

Mock-ups can be found in the attachments in projects zip file. Mock-up buttons involve functionality so you can use mock-up buttons to make the transitions among pages. Also you can find the screenshots of the prototypes.

## 4. Glossary

- 1- Admin: The person who is in charge of the Cakery App.
- 2- Cart: A basket where the items are added into.
- 3- Customized Cake: A cake which can be customized as buyer's wants and needs.
- 4- Dashboard: Home page of Cakery App.
- 5- Item: Cakes offered in bakeries.

- 6- User: Person who orders from the Cakery App.
- 7- My Earnings: Displays total earnings of bakery.
- 8- Product: Cakes offered in bakeries.
- 9- Profile: Detail of information about the user.
- 10- Seller: Bakeries that are enrolled in Cakery application.

## 5. References

- 1- Kirti Bhandge, Tejas Shinde, Dheeraj Ingale, Neeraj Solanki, Reshma Totare, "A Proposed System for Touchpad Based Food Ordering System Using Android Application", International Journal of Advanced Research in Computer Science Technology
- 2- Resham Shinde, Priyanka Thakare, Neha Dhomne, Sushmita Sarkar, "Design and Implementation of Digital dining in Restaurants using Android", International Journal of Advance Research in Computer Science and Management Studies 2014.
- 3- Ashutosh Bhargave, Niranjana Jadhav, Apurva Joshi, Prachi Oke, S. R. Lahane, "Digital Ordering System for Restaurant Using Android", International Journal of Scientific and Research Publications 2013.
- 4- Khairunnisa K., "The application of wireless food ordering system", MASAUM Journal of Computing, Vol.1 Issue 2, Sept. 2009.
- 5- K.K. Reddy, B. Naresh, "Intelligent E-restaurant using Android OS", Int. J. of Scientific Engg. and tech. research Vol. 3, Issue 22, September 2014, pp: 4383-4385
- 6- Asan, N. Badariah in the research paper, "Zigbee-based smart ordering system(S O S)", International Journal of computer trends and technology (IJCTT) Volume 11 No-5, May 2014.
- 7- Patel Krishna, Patel Palak, Raj Nirali, Patel Lalit, "Automated Food Ordering System", International Journal of Engineering Research and Development (IJERD) 2015
- 8- Adithya. R., A. Singh, S. Pathan, and V. Kanade, "Online Food Ordering System," Int. J. Comput. Appl., vol. 180, no. 6, pp. 22–24, 2017.
- 9- 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(IJISSET) 2015.
- 10- Patel, Mayurkumar, "Online Food Order System for Restaurants" (2015). Technical Library. Paper 219.

