

## Foodygo

Use Case Specification: UC04– Foodygo - Login Corrective Improvement

---

### USE CASE SPECIFICATION

<b>Product Name</b>	Foodygo User Portal App
<b>Date</b>	07/02/2024
<b>System</b>	User Portal App
<b>Use Case Number and Name</b>	Login Corrective Improvement

<b>Version Number</b>	1.3
<b>Author</b>	Tarasita
<b>Product Owner</b>	Y.Hoo
<b>Key User</b>	Consumer

### Revision History

<b>Date</b>	<b>Version</b>	<b>Author</b>	<b>Reviewed By</b>	<b>Change Description</b>
20210601	1.1	Rey		Change logo Seven Butlers to 7B. See screen definitions  Remove login title, see yellow marks
20240626	1.2	Rey		Change logo 7B to foodygo. See screen definitions
02072024	1.3	Tarasita		<ul style="list-style-type: none"><li>- Description Correction</li><li>- Correct Alternative Flow</li><li>- Happy Flow Login Corrective Improvement</li></ul>

**Table of Contents**

1. Brief Description
  - 1.1. Goal
  - 1.2. Actor(s)
  - 1.3. Preconditions
  - 1.4. Post Conditions
  - 1.5. Related Use Cases
2. Basic Flow
3. Alternative Flows
4. Exception Flows
5. Functional Detailed Requirements
  - 5.1. Field definitions
  - 5.2. Screen definitions

# Foodygo

## Use Case Specification: UC04– Foodygo - Login Corrective Improvement

---

### 1. Brief Description

#### 1.1. Goal

*[Execution of the Use Case results in achieving a goal for the user. This section describes the goal or end result. A Use Case should be achievable in one user session, i.e. it describes a continuous interaction between the user (actor) and the system]*

By this use case the user is doing the login through email and google sign in. The goal is to login to foodygo system and get benefit of food inspiration by see videos, helping other users by upload media contents (videos), write an article, see restaurant directory, play games.

#### 1.2. Actor(s)

*[An actor initiates the use case. Often actors map to roles in the application. This section describes the actor(s) that interact with the system in this use case]*

Consumer: user who wants to find food inspiration by see videos, upload media content, share video, etc

#### 1.3. Preconditions

*[Preconditions need to be fulfilled before the Use Case can start. They can be associated both to conditions outside the system as well as to the status of the system itself. In both cases preconditions must be checked. Preconditions are described in a numbered list to enable referencing]*

1. Consumer has registered to Foodygo
2. Consumer has a Google account

#### 1.4. Post Conditions

*[Post conditions list the status of the system and/or fulfilled requirements after successful execution of the Use Case. Post conditions are described in a numbered list to enable referencing]*

1. Consumer can log in to the system

#### 1.5. Related Use Cases

*[Within a Use Case other Use Cases could be used. An extended Use Case is a Use Case that could also be executed by on its own. It provides additional functionality on top of the Use Case in this document. Included Use Cases describe scenarios that are shared between various Use Cases and therefore described in a separate document. Note: beware of functional decomposition by excessive use of included use cases]*

##### 1.5.1. Extended Use Cases

UC03 – Register

##### 1.5.2. Included Use Cases

Not applicable

## Foodygo

### Use Case Specification: UC04– Foodygo - Login Corrective Improvement

---

#### 2. Basic Flow

*[The basic flow describes the most common series of interactions between actor(s) and system to achieve the goal of this Use Case. Steps are numbered to enable referencing. For steps describing the system's response the actor is System. Detailed requirements shall be described in sections 5 and 6 and/or appendices.]*

The use case starts when the consumer is on the login page.

Step	Actor	Description
1	System	Shows the login page: <ul style="list-style-type: none"><li>• Log in with email</li><li>• Log in with Google</li></ul>
2	Consumer	Select “Log in with email”
3	System	Show login page: <ul style="list-style-type: none"><li>• Fields Email address</li><li>• Fields password</li><li>• “Forgot password” button</li><li>• “Log in” button</li><li>• Don’t have account? Register (link)</li></ul>
4	Consumer	Fills the fields Email Address and Password. Clicks on button “Login”
5	System	Validates email address fields and password are not empty
6	System	Validates email address format (name@provider.extension)
7	System	Validates if the email is existing and if the password belongs to the email
8	System	Opens Home page

The use case ends.

#### 3. Alternative Flows

*[The basic flow does not need to be the only way to achieve the goal. This part of the Use Case describes alternative paths to successfully achieve the end goal. Steps in alternative scenarios may be replacing or extending steps in the basic flow]*

This flow starts if user click log in with social media.

Step	Actor	Description
1	System	Shows the login page: <ul style="list-style-type: none"><li>• Log in with email</li><li>• Log in with Google</li></ul>
2	Consumer	Select “Log in with Google”
3	System	Show login page: <ul style="list-style-type: none"><li>• Log in with Google</li></ul>
4	Consumer	Follow the flow from Google
5	System	Validates user’s account
6	System	Opens Home page

#### 4. Exception Flows

*[Exception flows describe scenarios where the Use Case goal is not achieved. Only Use Case specific scenarios are described. Generic exception flows that apply to all Use Cases for this application or to all in-house developed applications should be described in respectively a general requirement or a 'Standards & Guidelines' document]*

##### 4.1. One of the mandatory fields is not filled.

*[Describe the situation when this flow will be applicable. Include a reference to a step in basic or another alternative flow that acts as the starting point of this alternative scenario. Also reference the step where the Use Case continues after successfully ending this alternative flow. The use case may also end with this alternative flow of course]*

This exception flow applies when a mandatory field is not filled.

Starting point(s): replacing basic flow from step 2

Step	Actor	Description
1	System	Shows the login page
2	Consumer	Fill the fields Email Address. The Password is empty.
3	Consumer	Click on button "Login"
4	System	Validates email address fields and password are not empty
5	System	Shows the text on the password fields: "Password is required"
6	System	Set focus at the first field which should be filled correctly.

The flow is also the same for every mandatory field that should be filled. In this case, the email address field is included.

##### 4.2. Email is not containing the right format.

This exception flow applies when email is not filled in the right format

Starting point(s): replacing basic flow from step 2.

Step	Actor	Description
1	System	Shows the login page
2	Consumer	Fill the fields Email Address (wrong format) and Password
3	Consumer	Click on button "Login"
4	System	Validates email address fields and password are not empty
5	System	Validates email address format (name@provider.extension)
6	System	Shows the text on the email address field: "This email contains incorrect format"
7	System	Set focus at the email field

##### 4.3. Email address does not exist.

This exception flow applies when email is does not exist

Starting point(s): replacing basic flow from step 2.

## Foodygo

### Use Case Specification: UC04– Foodygo - Login Corrective Improvement

Step	Actor	Description
1	System	Shows the login page
2	Consumer	Fill the fields Email Address and Password
3	Consumer	Click on button “Login”
4	System	Validates email address fields and password are not empty
5	System	Validates email address format (name@provider.extension)
6	System	Validates if the email is existing and not deleted.
7	System	Shows the text on the email address fields: “This email does not exist”.
8	System	Set focus at the email field

#### 4.4. Password is not valid

This exception flow applies when password is not valid.

Starting point(s): replacing basic flow from step 1.

Step	Actor	Description
1	System	Shows the login page
2	Consumer	Fills the fields Email Address and Password
3	Consumer	Clicks on button “Login”
4	System	Validates email address fields and password are not empty
5	System	Validates email address format (name@provider.extension)
6	System	Validates if the email is exist and if the password belongs to the email
7	System	Shows the text on the password fields: “This password is not valid. you have x retry left”.
8	System	Set focus at the field confirm password

User can only retry for Password for 5 times. When user have retried for 5 times, the account is locked. The retry will be reset after 30 minutes.

#### 4.5. Account is locked.

This exception flow applies when user have tried for 5 times, and account is locked for 30 minutes.

Starting point(s): replacing basic flow from step 1.

Step	Actor	Description
1	System	Shows the login page
2	Consumer	Fills the fields Email Address and Password
3	Consumer	Clicks on button “Login”
4	System	Validates email address fields and password are not empty
5	System	Validates email address format (name@provider.extension)
6	System	Validates if the email is exist and if account is not locked.
7	System	Shows the text on the password fields: “Account is locked. Please try again in 30 minutes”.
8	System	Set focus at the field confirm password

## Foodygo

### Use Case Specification: UC04– Foodygo - Login Corrective Improvement

---

If it's already 30 minutes after locked, Unlock the account and check the password.

#### 5. Functional Detailed Requirements

*[This section allows for detailed specifications of steps mentioned in the flows above. This may include process logic, business logic, validations, screen definitions etc.]*


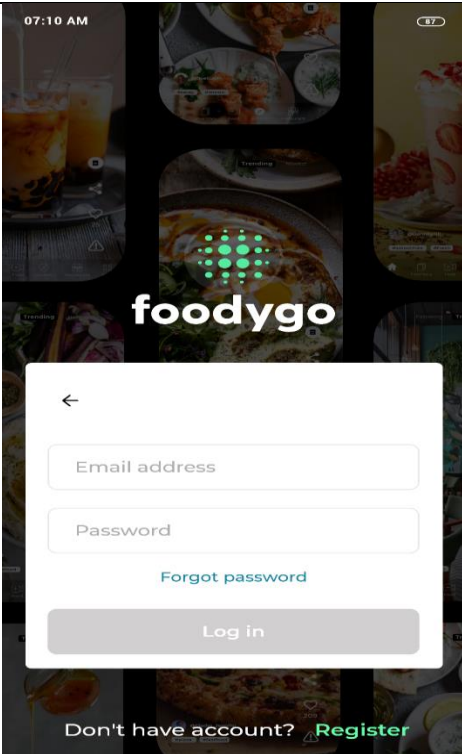
##### 5.1. Field definitions

Field	Mandatory	Max length	Validation
Email Address	yes		Email format
Password	yes		

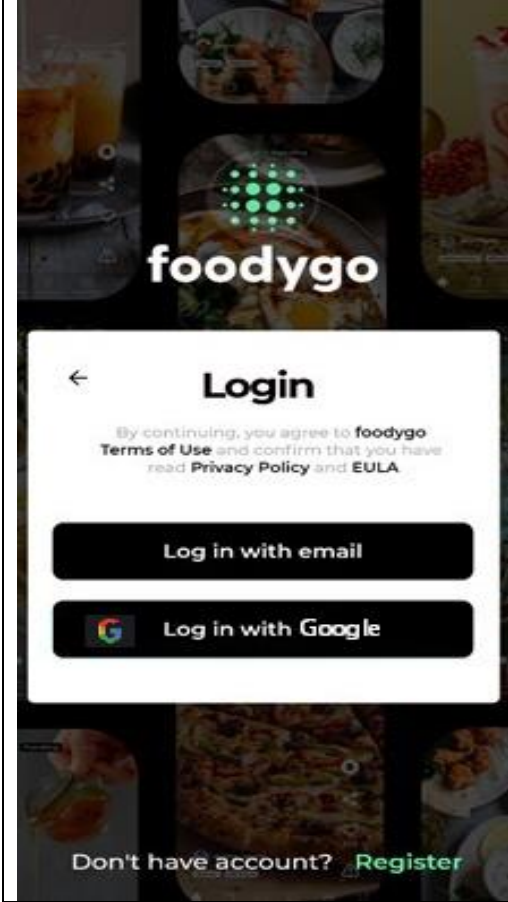

##### 5.2 Screen definitions

First page is the Login page where consumer can fill email address and password that they registered before.

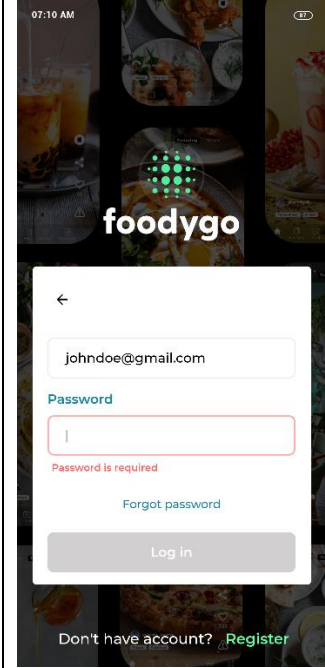
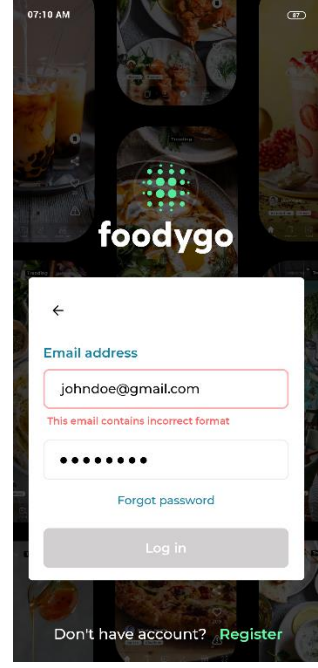
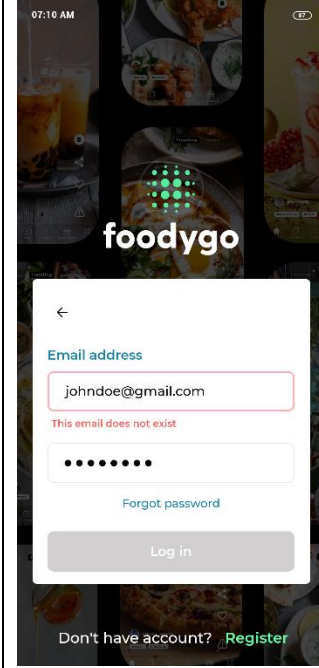
Left image below is the login page and the right image shows the red text when the input field is empty or contains wrong value.

Step 1-2 Log in with email	Log in with email page
	

Log in with Google

Log in with iOS	Log in with Android
	

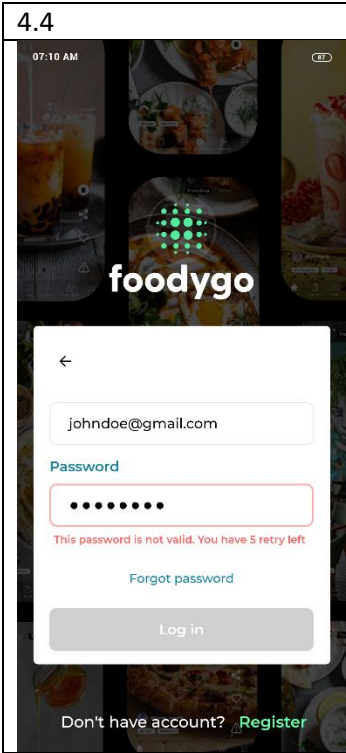
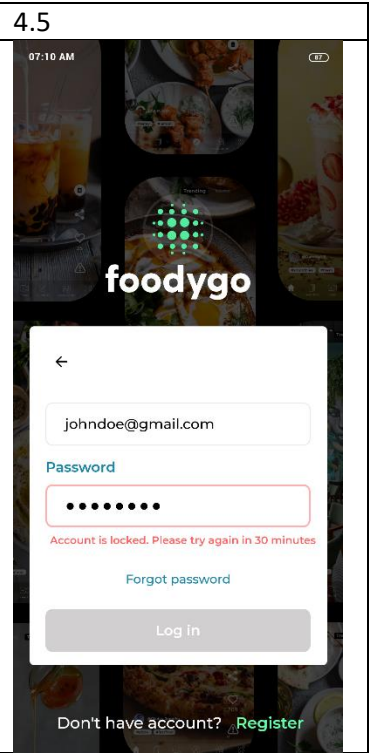
Exception Flows

4.1	4.2	4.3
		



Foodygo

Use Case Specification: UC04– Foodygo - Login Corrective Improvement

4.4	4.5
 <p>This screenshot shows the Foodygo login interface. At the top, the status bar displays '07:10 AM' and a battery icon. The background features a collage of food images with the 'foodygo' logo in the center. A white login form is overlaid, containing a back arrow, an email input field with 'johndoe@gmail.com', a password input field with masked characters, and a red border around the password field. Below the password field, a red error message reads: 'This password is not valid. You have 5 retry left'. A blue link 'Forgot password' is positioned below the error message. At the bottom of the form is a grey 'Log in' button. Below the form, the text 'Don't have account? Register' is displayed, with 'Register' in green.</p>	 <p>This screenshot shows the Foodygo login interface after a failed login attempt. The layout is identical to the previous screen, but the error message below the password field is: 'Account is locked. Please try again in 30 minutes'. The 'Forgot password' link and 'Log in' button remain visible. The 'Don't have account? Register' text is also present at the bottom.</p>