

University for Applied Sciences
Informatics Department
Applied Informatics

No More Waste

Documentation for the Architecture of an Mobile Application for Preventing Food
Waste

Bruno Macedo da Silva
676839
inf3645@hs-worms.de

Supervisor	Prof. Dr. Volker Schwarzer
Working Period:	Summer Semester 2022
Due Date:	31.Juni 2022

Contents

1	Introduction and Goals	3
1.1	Design Purpose	4
1.2	Functional Requirements	4
1.3	Quality Goals	6
1.4	Stakeholders	7
2	Constraints	8
3	Context and Scope	10
3.1	Business Context	10
3.2	Technical Context	11
4	Building Block View	12
4.1	Scenario view	12
4.2	Structural view	13
4.3	Behavior view	16
5	Crosscutting Concept	17
5.1	Solution for Usability	17
5.2	Solution for Interoperability	17
5.2.1	Payment Gateway	18
5.2.2	Federated Authentication	19
5.3	Solution for Performance	19
5.4	Solution for Security	20
6	Quality Requirements	21
6.1	Quality Tree	21
6.2	Evaluation Scenarios	22
7	Risk and Technical Debt	26
7.1	Motivation	27
	References	29
	Abbreviations	31

1 Introduction and Goals

According to the Food and Agriculture Organization of the United Nations (FAO) in 2019, 931 millions tonne of food were wasted [FAO (2013)]. This has environmental, but especially social consequences. In a world where approximately 9.9% of the [AAH (2022)] population suffers from hunger that waste percentage sounds paradoxical.

According to United Nations (UN) 5% of the global food loss and waste comes from restaurants [UN (2022)]. The solution for this problem must be locally applied so its effects can be seen in a global structure. To do so we propose to develop a mobile application that connects restaurants, bakeries and or pastries to clients. The former would offer their remaining products, which are still consumable, prior to the closing time, to a small price and the latter would browser in the app to find which shops are offering products.

We as “Clean Up the World ®” are a rising StartUp whose main concerns is to find environmental solutions to daily problems. Our portfolio includes projects about management of waste and optimization of household water usage. This product we want to develop targets small communities, like small cities or regions within a big city, to reduce the amount of wasted consumable food.

With our project we want to achieve the following goals:

- Connect providers with clients, so the former can offer products that the latter can purchase
- Collect statistical data about waste reduction within the providers
- Promote reduction of food waste that still could be consumed
- Allow clients to have a different dining experience.
- Allow providers to promote their products and gather new clients.

To make the easy to read we will use the pronouns “‘he”’ and “his” every time we refer to a single person.

1.1 Design Purpose

The main purpose of this architecture is creating an exploratory prototype of an App. We aim to test it with potential stakeholders and regions to analyze their general acceptance and wishes [Cervantes and Kazman (2016)] and get a fast feedback.

This prototype will also make it feasible to identify unknown needs and wishes of the potential stakeholders, so we can eventually increase the scope of functionality. Exploring this domain will also provide us with information regarding the behavior of our target group when it comes to buying and serving food that would be wasted, but is still consumable.

1.2 Functional Requirements

The following functionalities describe the basic requirement for the App:

Id	Requirement	Description
F-1	Register as Client.	A Client can register to the app with its e-mail.
F-2	Login	After registration Client can login into the app.
F-3	Purchase option	A registered Client can purchase an available offer (see F7).
F-4	Filter/search options	A Client can perform filter and search actions for products.
F-5	Register as Provider	A Provider can register his store and add logos and pictures.
F-6	Create offer	A registered Provider can publishes what products they are offering with price and amount.
F-7	Upload offer	A registered Provider can add, edit or remove offers to his catalog.
F-8	Check orders	A registered Provider can check all existing orders targeting his/her shops.

ID	Motivation
F-1	The entry door of the App, where our Client get an overview of all available offers
F-2	In order to place purchases our client need to be registered. It will also provide statistical information about consumer behavior
F-3	Since we are dealing with a business relationship we have on one side a client willing to pay and for a product and on the other side a provider willing to offer a product/service
F-4	Like any other online-shop it is important that our Client can browse through the available possibilities
F-5	In order to make a product available a Provider needs to register his/her shop. This information will also be used for statistical analyzes about providers, products and consumer behavior
F-6 - F-7	A registered Provider can make an offer available according to his/her daily planning. For future development of this app, this will be helpful to identify tendencies regarding dates, periods and availabilities.
F-8	Also registered providers can get an overview about how often their products have been sold. This may open a different kind of business orientation.

36 The following Use Case Diagram should give our stakeholders an overview of the primary
37 functionality of the app:

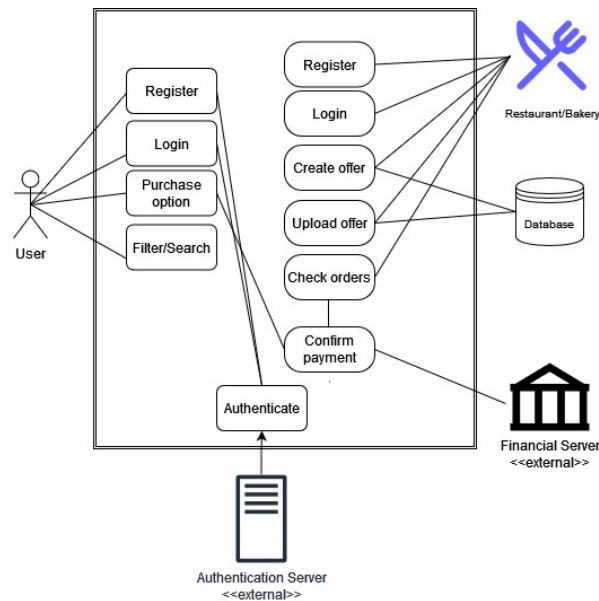


Figure 1: Preliminary functions

38 1.3 Quality Goals

39 The key qualities of this app are described in the table below:

Quality	Priority	Motivation
Usability	1	Since we are working with a prototype it is important the usage is easy as possible, to attract more users and to gather information about consumer behavior. clients and providers should have a simple interface where they can quickly interact without any burdens.
Interoperability	2	To reduce programming burdens and accelerate the delivery of a working product the registration and payment process will rely on third party providers. For that reason the developed features should work faultless in combination with the external Application Programming Interface (API) (i.g Mobile Payment Gateway and Federated Login).
Performance	3	Many mobile and web-apps lose potential users because of the lack of performance. A System Response that takes too long (more than 1 second [AppDynamics (2020)]) may frustrate potential users and discourage them of using the application.
Security	4	To guarantee a secure and easy payment process we will handle the API of the Mobile Payment Gateway within the development process. The possibility of outsourcing this service would cause a big damage to the first priority.

40 1.4 Stakeholders

41 The main stakeholders of this app are described in the table below:

Stakeholder	Description	Motivation
Providers	Owner of a restaurant, bakery or pastry.	One of the protagonist of this app. They will interact with clients using the app. From his usage we will gather valuable information about consumer behavior.
Clients	Person who wants to purchase last minute product from a provider.	The second protagonist of the app they will interact with the provider to search and to purchase product. The result of this interaction will provide us with statistical information to understand how food waste can be reduced.
Developers	Team in charge of creating the application using existing tactics and creating new solutions.	Responsible for guarantee that the main requirements of the app are fulfilled and fully functional. Since they will be dealing with the background of the product, it is important that they understand it very good so it can also be implemented in a final version.
Boarding Committee of "Clean Up the Word (R)"	Members of the management team who wants to delivery environmental solution do daily problems and at the same time develop a profitable product.	Group in charge of main decisions regarding what will be developed. Their decision are based on mark tendencies and on environmental issues.

2 Constraints

In this project we must distinguish between Technical (CT-T-#) and Business (CT-B-#) Constraints. The former describes specific elements of the project, like programming language, released platform (e.g. operational systems) and technical decisions related to the functionalities. The latter deals with management elements [Franzen and Thoms (2020)] (e.g time, budget and team). The following tables describes the technical and the business constraints of this project:

Technical		
Id	Constraint	Reasoning
CT-T-1	Programming Language - Kotlin	A multilanguage (Java, Kotlin, iOS, Swift) approach increases the maintainability burden and consequently the costs (see CT-B-4). It can also interfere with compatibility with different kind of device.s
CT-T-2	Platform - Android	Offering the application for different platforms (iOS and/or Android) increases costs for maintainability and requires a bigger team. Since the prototype should run during the first year mainly to gather information about consumer behavior the costs in this test phase can increase rapidly if we decide to develop for the most common platforms.
CT-T-3	Payment	One the one hand creating an own payment framework can gives full control of the application, but on the other hand it will required specialized team and increases costs and time (see CT-B-4).
CT-T-4	Payment gateway	Using existing Mobile Payment Gateway reduces development time, but demands fully Interoperability of the app with the existing gateways. It may also be a problem if the Client don't use this kind of payment method.
CT-T-5	Login	Using existing Federated Login decreases development time, but like CT-T-4 demands fully interoperability of the app with appliances. It may also be a problem if the Client don't trust this kind of login.
CT-T-6	Amount of server	The existing budget allow us to have only one full operating server to handle the input data.

Business		
Id	Constraint	Reasoning
CT-B-1	Time to first prototype release - Maximal 1 year	To stay on the budget the first release should be ready within one year after the approval
CT-B-2	Development Team - 4 Dev + 1 Sys-Admin	Most of the development team members is allocated on other projects and cannot be changed. They should be in charge of the development and maintainability.
CT-B-3	Analytical Team - 1 person	During running phase of the prototype it will be necessary to have someone in charge of evaluating and interpreting the collected data, to find out if the goals are being achieved.
CT-B-4	Budget	For this kind of project the maximum budget is US\$ 150.000. It should cover the development of the main functionality and the data analysis (see CT-B-3). “middle app” according to [SPD LOAD (2019)]
CT-B-5	Security and Trust of third party resources	Since login and payment will be processed by third party providers, Federated Login and API Gateway, a lack of security on their side can cause damage to the reputation of our company.

3 Context and Scope

Since this system relies on the correct working of external elements it is important that their interaction is correctly displayed.

3.1 Business Context

This graphic is addressed to the following stakeholders: providers, clients and Boarding Committee of “Clean Up the Word (R)”. It should give a level 1 view of the interaction the app and its external components.

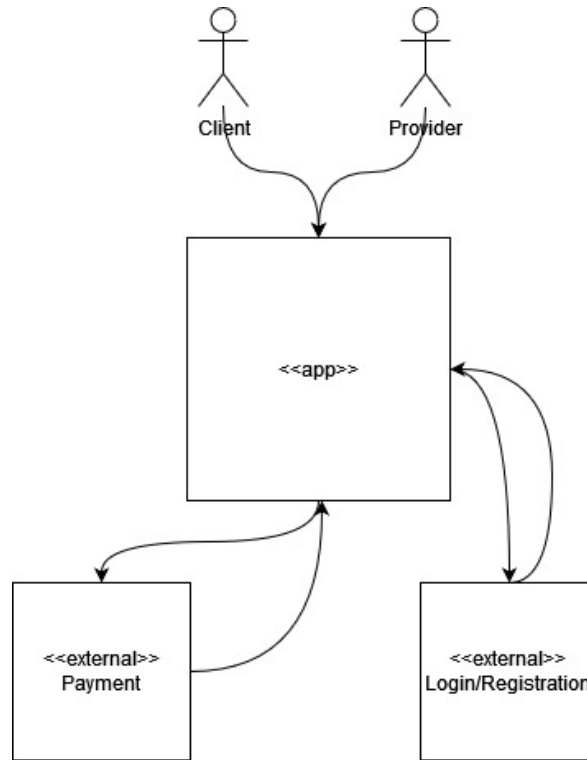


Figure 2: Diagram to describe the business context

Artefact	Description
Client	Searches for a last time offer from a restaurant, bakery or pastry.
Provider	Offers a still consumable product that was not sold during normal working time.
Payment	Deals with the payment processing using registered information from another payment platforms.
Login/Registration	Authenticated users using logins from other platforms.

3.2 Technical Context

This following graphic is addressed to our technical team. It provides a level 2 view of the previous description 4.

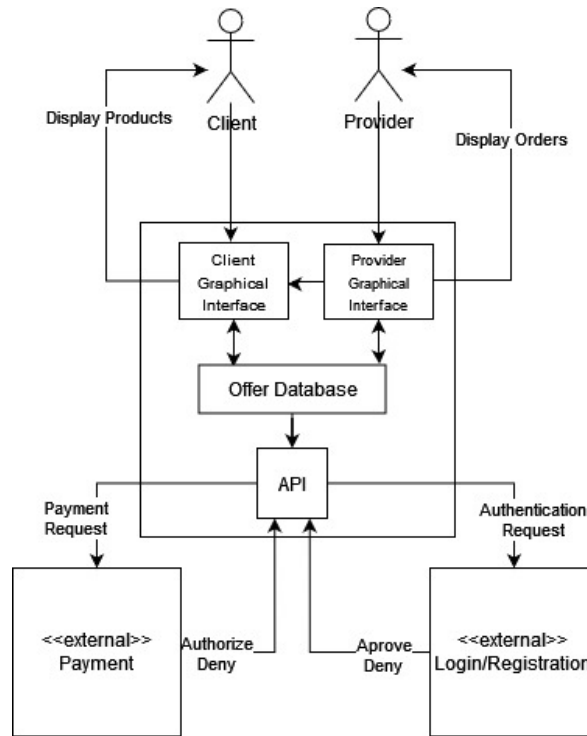


Figure 3: Technical Context

Artefact	Description
Graphical Interface	Client and Provider have an own interface to interact. Provider can access view their offer also with a Client's perspective.
Offer Database	Clients and providers can make requests to the database to inquire about its content.
API	For login and payment the authentication and authorization take places on the external service.

59 4 Building Block View

60 In this section we will describe the App using some elements of the 4+1 Architectural View
61 Model. With this model we aim to target an understanding of all our main stakeholders.

62 We will use five different views, which should focus on specific elements of the project. Each
63 view provides a different purpose Kruchten (1995). For this project we will provide the 3
64 following views of the 4+1 Architectural View Model:

- 65 • **Scenario view:** simple description for the end user
- 66 • **Structural view:** object-oriented decomposition
- 67 • **Behavior view:** description of the existing processes

68 4.1 Scenario view

69 Our first picture 1 provided our stakeholders a brief presentation of the basic functionalities
70 of our app. The following picture should give them a shallow view of the interaction the app
71 and its external components.

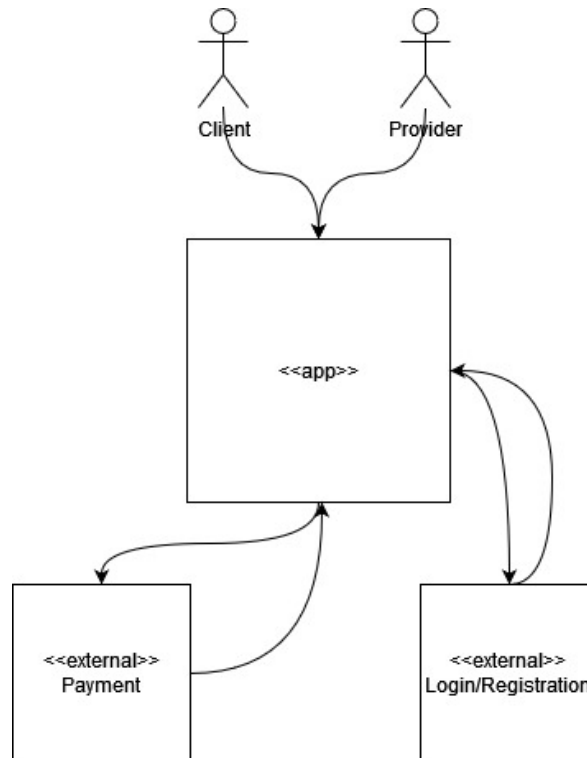


Figure 4: Diagram to describe the business context

Artefact	Description
Client	Searches for a last time offer from a restaurant, bakery or pastry.
Provider	Offers a still consumable product that was not sold during normal working time.
Payment	Deals with the payment processing using registered information from another payment platforms.
Login/Registration	Authenticated users using logins from other platforms.

More elements of this view will be presented while discussing the internal decisions in chapter 5.

4.2 Structural view

This following graphics are addressed to our technical team. It provides a deeper view of the model shown in 4. The once black-boxed elements showed above are now white-boxed so our development team becomes a better understanding of the relevant components of the app.

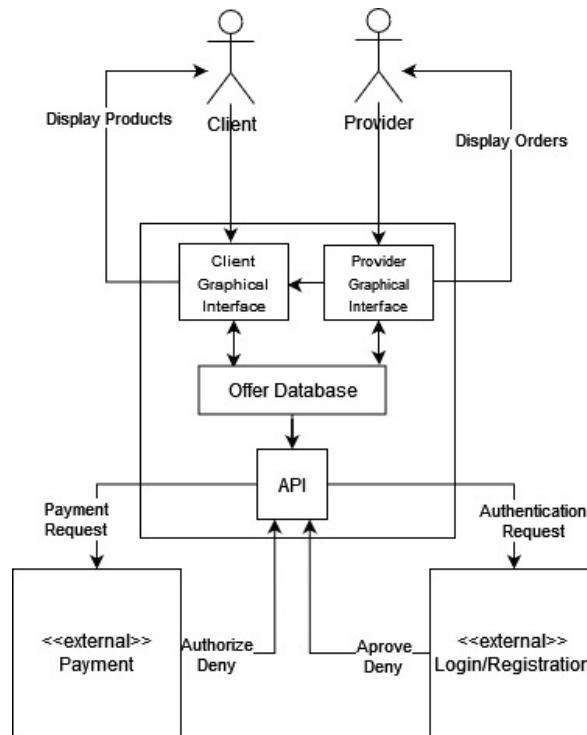
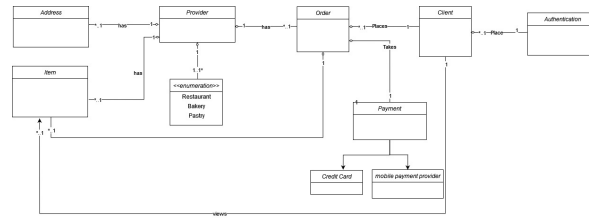


Figure 5: Technical Context

We choose a Class Diagram to provide our our development team a further view of the structural elements of the project. It canso help our other stakeholders to understand what are al



81 This final graphic show the whole classes in combination:

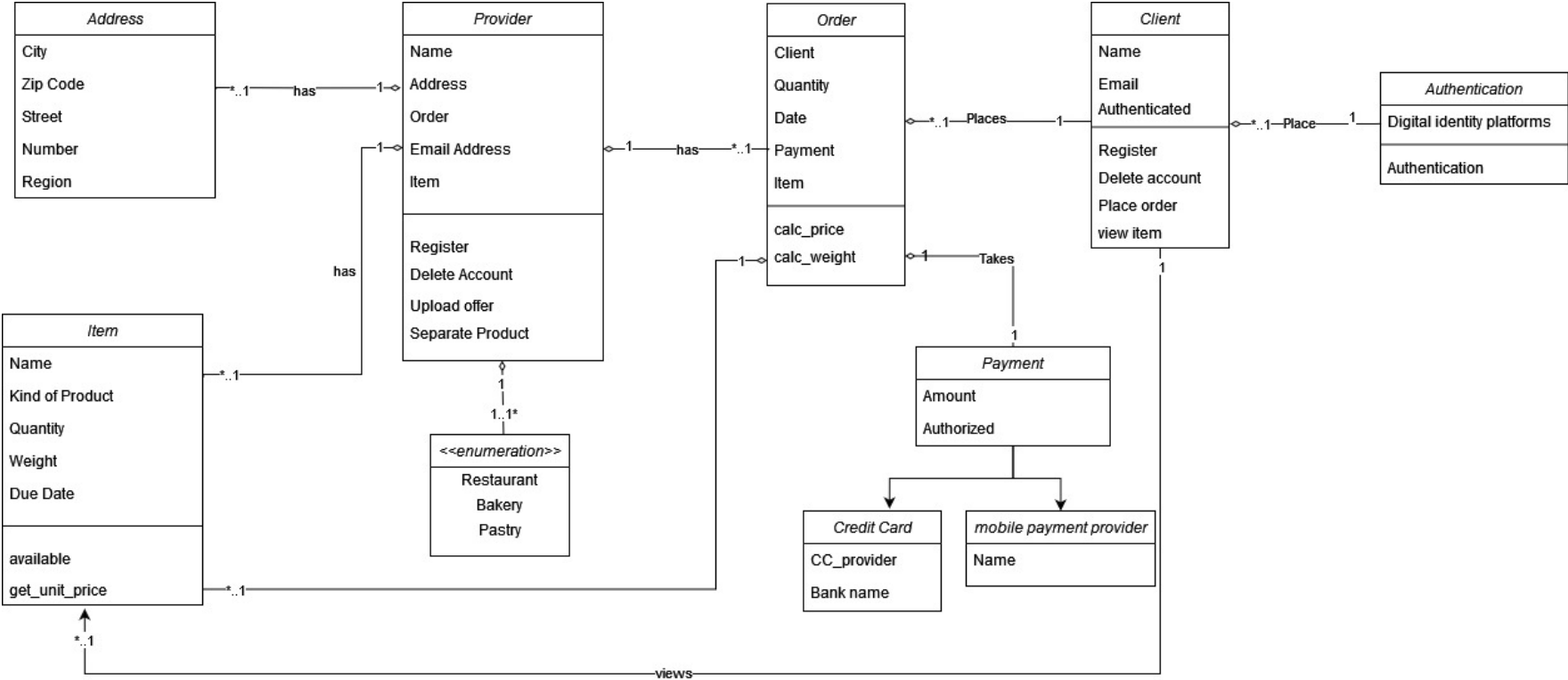


Figure 7: Classes Overview

82 4.3 Behavior view

83 The following Activity Diagram depicts the register and login procedure within the app. It
 84 should explain our main stakeholders, providers and clients, the starting process of the app.

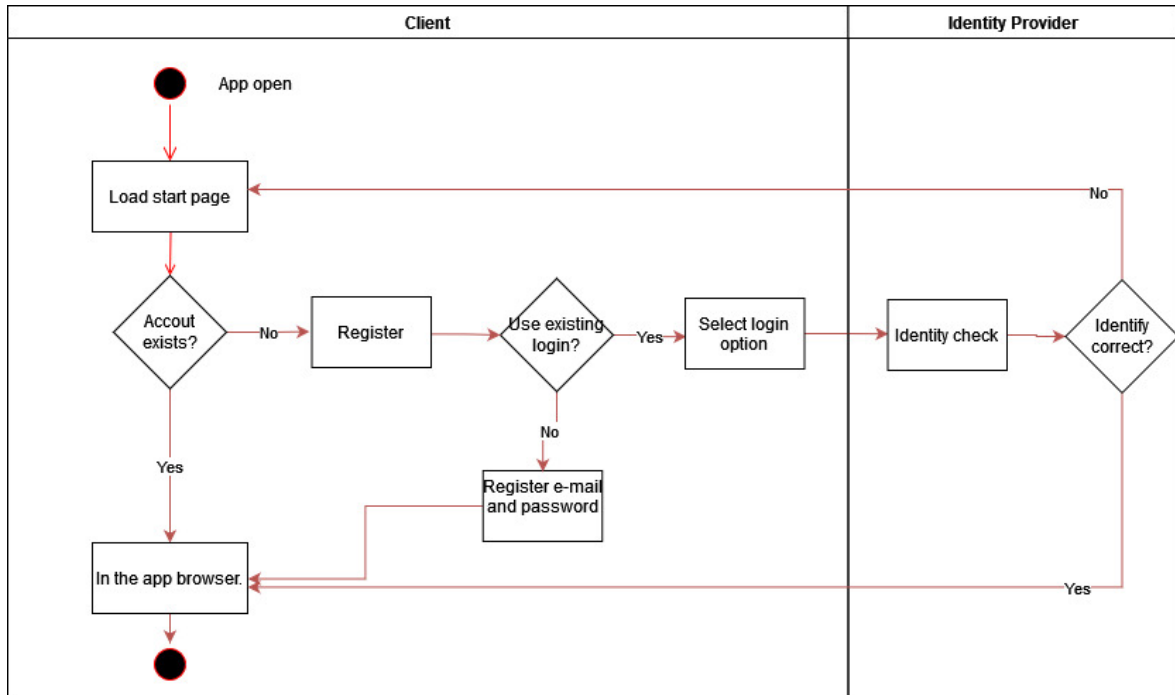


Figure 8: Login procedures

5 Crosscutting Concept

In this chapter we will present the technical solutions that we will use to develop this project. For each quality attribute we will present the chosen tactics.

5.1 Solution for Usability

The core of our app is how easy it is to use. We want our user to navigate through it without being overwhelmed with information not related to the main objective: purchase a product or upload a product.

Tactict	Pattern	Motivation	QA
Support User Initiative	Observer	The interaction of the users is a main factor of our app. We want them to have fully control of their actions either by cancelling or by resuming an action.	QA-1
	Lazy Registration	Avoid having to memorize another password and username may increase the acceptance of the user. With this pattern we allow them also to browse in the app and seeing what is available without being registered Interaction Design (2017). This may give a glimpse of what they get if they join us.	
Support System Initiative	Observer	By each upload from the providers we want our clients to have it on his device, without having to "ask" for it.	

5.2 Solution for Interoperability

The communication with the third party components should during the whole lifetime of the App reliable. Since we are dealing with two different services, Mobile Payment Gateway and Federated Login, we will describe the integration processes according to each specification.

From the third party applications we expect the following interaction:

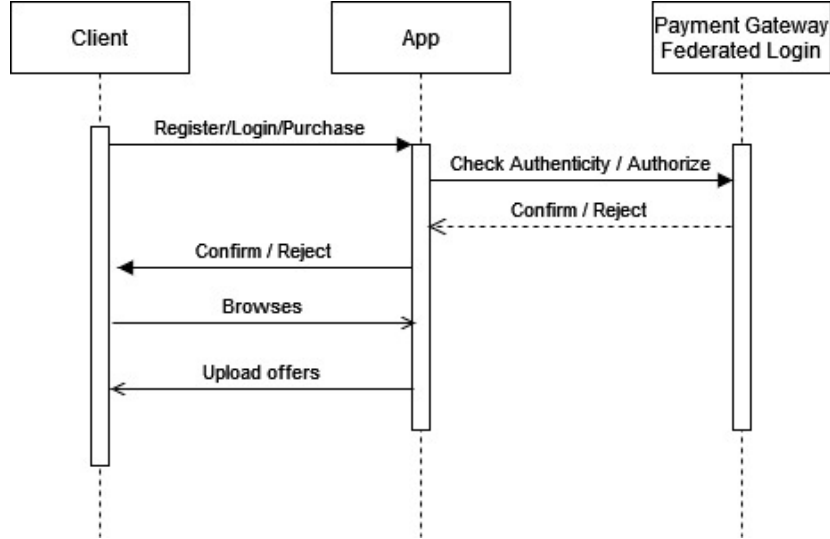


Figure 9: Sequence of actions with third party applications

5.2.1 Payment Gateway

The usage of Mobile Payment Gateway offers three possibilities Zoho (2019):

- Redirection to payment processor's page
- Payment data and processing inside the application
- Payment data entered in the app, but processed with an API

The third option stays in direct contact with our top quality attribute, usability. Since we want to offer a easy shopping experience, the payment process should also be harmonic with other features.

Tactict	Pattern	Motivation	QA
Limit Dependencies	Wrapper	The API will be the intermediary for the payment process. For the clients all visible steps will occur in the app, without being sent to another page. On the background the API will receive the input and send it to the payment gateway. The verification takes place in gateway, which then communicate with the financial institute of the client and send the payment to the Provider Zoho (2019).	QA-2

5.2.2 Federated Authentication

Using of Federated Login reduces burden of saving user credentials locally. It also improves the Usability so users do not have to create and remember another username and password. The authentication process takes place on the third party operator, as seen in the picture 9.

Tactict	Pattern	Motivation	
Microservice	API Gateway	Increase of security, so the microservice is not directly exposed to the external world. It reduces the complexity of the microservice, since the gateway will have to deal with data transfer rate, tokens and other activities. Dealing with failures would also be handled and logged by the microservice javarevisted (2021).	QA-2

5.3 Solution for Performance

We want our app to have a fast (no more than 1 second) response time. By clicking on an offer a Client should have it immediately displayed on his screen. Updated made by providers should also be promptly available for clients to browse.

Tactics	Pattern	Motivation	QA
Increase Resources	Load Balancer	This maybe implemented if, during the first test phase, we see that the usage of the app is so high that the existing components become overwhelmed. Specially during peak times we want our users to have a smoothly and fast interaction with the app. Providers and clients should perform their tasks, either browsing, purchasing or uploading offering without having to wait to get a response. With this decision all requests would be forwarded to the server that are available avoiding queuing of requests.	QA-3

113 5.4 Solution for Security

114 There are two security concerns that need to be addressed to the users. The first one deals with
 115 the authentication and payment process. This will be managed by the third party providers.
 116 The second one involves the interaction of the providers with the app. Since this stakeholder
 117 can upload data and file to the app it is important that only approved data type is inserted.
 118 In the table below we will describe the tactics used for the these two concerns.

Tactic	Pattern	Motivation	QA
Validate Input	Intercepting Validator	providers has a big interaction with the app. They can upload files and texts. To make sure that only secure element a inserted into the app, it is important that every input is analyzed before reaching the app and the clients Steel et al. (2012).	QA-4
Authenticate Actors Authorize Actors	Authentication enforcer Authorization enforcer	To avoid the connection of bots we want to allow only registered users to interact with the functionalities of the app. This will be done with the third party operators [Wikipedia (2020)].	QA-4

119 6 Quality Requirements

120 6.1 Quality Tree

121 The priority of each element will be expressed using the following notation:

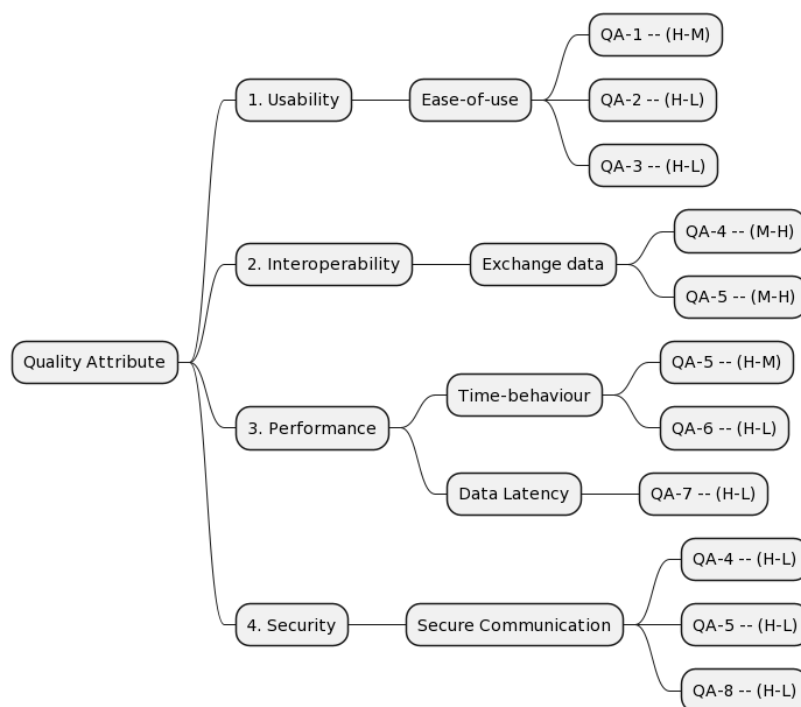
122 • ([Customer view], [Architect view])

123 • H - High

124 • M - Medium

125 • L - Low

126 Users and the development team have different perspective of an app. The former think about
127 how attractive and easy to use it is, the latter want to build something what achieves a goal.
128 For that reason is the interpretation of the priority sometimes so different, according to which
129 group has been asked.



ID	For Users	For Development Team
QA-1	All important information should be there so his shop can be well promoted	An initial registration with filter for the input is important, but aesthetically details are the goal now.
QA-2	Once he get something new, he wants to make it available	First we need to guarantee that no overrides occur than they see if it is promptly displayed.
QA-3	They want to browse and see all available options	Search engine can be very helpful, but filtering can wait a little, since it does not affect the app itself
QA-4	The most important is that they can use and purchase	This integration must be done fast and careful so no mistakes shows up.
QA-5	They just want to easily and secure pay, it does not matter how it works	The compliance with payment regulations is a must, since any mistake can costs huge fines and damage to the image of the company.
QA-6	They don't want to waste time with loading pages	The loading time can be fixed once there is a structure that allows loading in the first place.
QA-7	They want to get confirmation that everything worked fine.	The communication between the API and the payment provider show comply with all existing regulations. Push notification can be added once the main feature works.
QA-8	That is something that they don't want to see, but want to make sure that it exists	Since the payment is processed by the third party operator, all concerns should be addressed to them and specified in the Service Level Agreement (SLA)

130 6.2 Evaluation Scenarios

131 From the requirements, 1.2, we could develop the following uses cases and depict the main
132 quality attributes of this project.

Use Case	Description
UC-1: Register as Client	The Client registers an e-mail address.
UC-2: Login	The Client logs in to the system.
UC-3: Places an order	The Client chooses a Provider.
UC-4: Register payment	The Client registers a payment method.
UC-5: Register as Provider	The Provider registers their facility and products.
UC-6: Update availability	The Provider uploads their product catalog.

133 With the following use cases we will be able to define the major quality attributes that are
134 involved in the development of this application. They should be measurable and testable so

135 we can verify if the system meets the needs our stakeholders [Cervantes and Kazman (2016)].

ID	Quality Attribute	Scenario	Associated Use Case
QA-1	Usability	A Provider is able to register his company, to specify the kind of products he offers and upload a logo or picture of his shop and products in a easy and fast (within 5 Minutes) fashion.	UC-5
QA-2	Usability	A Provider is able to update the offers at any time.	UC-6
QA-3	Usability	A Client is able to search and filter options.	UC-6
QA-4	Interoperability	A Client can register his e-mail using another account (Google, Microsoft, Facebook) in a Federated Login	UC-1
QA-5	Interoperability	A Client can pay the order using a Mobile Payment Gateway (e.g. Stripe, Square, PayPal, SecurePay)	UC-4
QA-5	Performance	A Client registers his/her e-mail address and can immediately browse in the app.	UC-1
QA-6	Performance	A Client opens the app and he can immediately search for products or providers.	UC-2
QA-7	Performance	A Client chooses a Provider and places his order. After the confirmation of payment, a push-message is displayed in the app confirming the purchase.	UC-3
QA-8	Security	The payment process should be secure and within the app. It should also give the Client the feeling of security. The Client inserts his payment information it is processed by the payment operator.	UC-4 & QA-5

136 The defined quality attributes are represented in the following scenarios:

Usability			
Scenario	Value		
Source	Provider	Registered Provider	Client
Stimulus	wants to register his shops	wants to make a last minute offer	wants to search/filter offers
Artifact	app	app	app
Environment	working time, during afternoon	peak period, between 4 and 7 pm on Friday	peak period, between 4 and 7 pm on Friday
Response	offer available in the app	immediate availability of the offer in the app	display of the filter/search output
Response Measure	How long did the registration and upload process take? How many and what kind of error messages did the Provider get?	How long did it take to upload an offer? How many and what kind of error messages did the Provider get?	What kind of inputs did the user has to place until he finds what he wants? Did he have to type anything or were filter/search options available? How long it takes until the client finds a product?

Interoperability		
Scenario	Value	
Source	Client	Client
Stimulus	wants register using a Federated Login	wants to pay using existing mobile payment account
Artifact	app and Federated Login provider	app and Mobile Payment Gateway
Environment	peak period (on the context of the Federated Login provider)	peak period (on the context of the gateway)
Response	authentication succeed or failed	confirmation / declined
Response Measure	How much data was transmitted and how much was queued? Focus on System overload [Kasunic and Anderson (2004)]	Total amount generated data in the app that are transferred and processed and rejected by the gateway? Focus o connectivity and system overload [Kasunic and Anderson (2004)]

Performance			
Scenario	Value		
Source	Client	Client	Client
Stimulus	wishes to create an account	wants to search for a Provider	places an order
Artifact	app	app	app
Environment	weekend between 3 and 7 PM	peak period, between 6 and 7 pm on a Friday	peak period, between 6 and 7 pm on a Friday
Response	immediate access to the app	immediate access to the offers	confirmation of payment / payment declined
Response Measure	time between confirmation and access	how quickly does the client's device get update of availabilities	How long did take until the client get the confirmation/declined of payment?

Security		
Scenario	Value	
Source	Client	Client
Stimulus	clicks on registration using an existing login	click on pay using an existing mobile payment account
Artifact	app, API Gateway and Federated Login provider	app, Microservice and Mobile Payment Gateway
Environment	peak period (on the context of the Federated Login provider)	peak period (on the context of the gateway)
Response	authentication succeed or failed	confirmation / declined
Response Measure	Required time and effort to intercept and/or block requests (create Denial of Service (DoS))	Extension to image damage of the app and of the company in case of attack

137 7 Risk and Technical Debt

138 To measure the risks of this project we will use the following 3x3 risk matrix, which will help
 139 us develop the Risk Assessment:

3x3 RISK MATRIX

		SEVERITY →		
	LIKELIHOOD ↓	1	2	3
1		LOW - 1 -	LOW - 2 -	MEDIUM - 3 -
2		LOW - 2 -	MEDIUM - 4 -	HIGH - 6 -
3		MEDIUM - 3 -	HIGH - 6 -	HIGH - 9 -

Figure 10: 3x3 Risk Matrix Template
 Source: [Smartsheet (2017)]

140 On the left side we see the risk table defined after several discussion with the team members.
 141 On the right side there are the elements to which the table refers to:

Risk Criteria	Element ID					
	1	2	3	4	5	6
Unproven technology	1	1	1	1	1	1
Performance	2	2	1	1	1	1
Scalability	1	1	2	1	1	1
Availability	1	1	4	2	1	1
Data loss	1	1	3	1	1	1
Single points of failure	1	1	4	4	1	1
Security	1	2	3	2	2	2

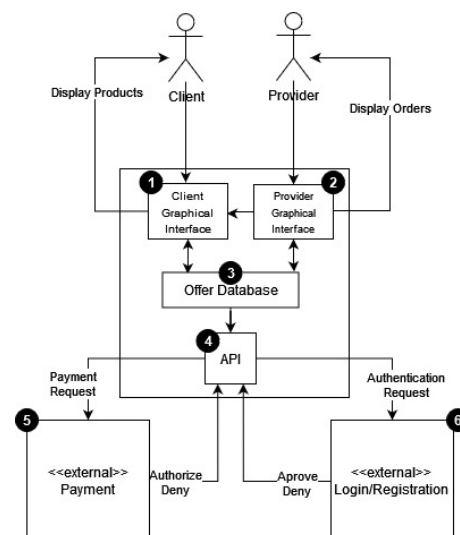


Figure 11: Modified from Figure 5

Risk Criteria	1 Client Graphical Interface	2 Provider Graphical Interface	3 Offer Database
Unproven technology	<i>not applicable</i>	<i>not applicable</i>	<i>not applicable</i>
Performance	Concerns regarding a such dynamic shop. Data displayed	Concerns regarding a such dynamic shop. Data update.	<i>not applicable</i>
Scalability	<i>not applicable</i>	<i>not applicable</i>	One database can be overwhelmed if the number of user is not limited for this prototype version.
Availability	<i>not applicable</i>	<i>not applicable</i>	In case of intense traffic the latency can be more than expected.
Data loss	<i>not applicable</i>	<i>not applicable</i>	Nowadays no company can survive in case of data loss. Technical damages can be mostly easy fixed, but moral damage stays forever.
Single points of failure	<i>not applicable</i>	<i>not applicable</i>	Only one component gives always this risk. Increasing the number of components increases also the total development costs
Security	<i>not applicable</i> the client's input is always processed by the APIs and by the third party providers	If no strong data filtering exists, providers can upload malicious files or execute unwished commands.	The filtering of the input should occur mostly on the server side, Source no external access can figure out, how it is implemented.

Risk Criteria	4 API	5 External Payment Service	6 External Authentication Service
Unproven technology	<i>not applicable</i>	<i>not applicable</i>	<i>not applicable</i>
Performance	<i>not applicable</i>	<i>not applicable</i>	<i>not applicable</i>
Scalability	<i>not applicable</i>	<i>not applicable</i>	<i>not applicable</i>
Availability	The access to the products become unstable.	<i>not applicable</i>	<i>not applicable</i>
Data loss	<i>not applicable</i>	<i>not applicable</i>	<i>not applicable</i>
Single points of failure	In case of failure login, registration and payment are compromised.	<i>not applicable</i>	<i>not applicable</i>
Security	Lack of practical experience within the team about this topic. we rely on the service provided by the third party companies	SLA Less than 99.5% but equal to or greater than 95.0% [Paycore (2019)]	SLA < 99.99% - >= 99.9% auth0 (2014)

143 References

- 144 AAH (2022). World hunger: Key facts and statistics 2022. *actionagainsthunger.org*. <https://www.actionagainsthunger.org/world-hunger-facts-statistics>, Accessed 18th May
145 2022.
146
- 147 Amplify Docs (2020). Federated identities. <https://docs.amplify.aws/sdk/auth/federated-identities/q/platform/android/>. Accessed 9 June 2022.
148
- 149 AppDynamics (2020). 16 metrics to ensure mobile app success. <https://www.hwe.design/theories-concepts/system-response-stability>. Accessed 8th June 2022.
150
- 151 auth0 (2014). Service levels. <https://www.smartsheet.com/all-risk-assessment-matrix-templates-you-need>. Accessed 10th June 2022.
152
- 153 AWS (2017). Microservices. https://aws.amazon.com/microservices/?nc1=h_ls. Accessed
154 9th June 2022.
- 155 Baresi, L. (2009). *Activity Diagrams*, pages 41–45. Springer US, Boston, MA. https://doi.org/10.1007/978-0-387-39940-9_9, Accessed 26th May 2022.
156
- 157 Cervantes, H. and Kazman, R. (2016). *Designing Software Architectures: A Practical Approach*. Pearson Education, Boston.
158
- 159 FAO (2013). Food wastage: Key facts and figures. *fao.org*. <https://www.fao.org/news/story/en/item/196402/icode/>, Accessed 18th May 2022.
160
- 161 FAO (2022). 17 *fao.org*. <https://www.fao.org/food-loss-reduction/news/detail/en/c/1378973/>, Accessed 18th May 2022.
162
- 163 Franzen, E. and Thoms, M. S. (2020). Architectural drivers in modern software architecture. <https://medium.com/@janerikfra/architectural-drivers-in-modern-software-architecture-cb7a42527bf2>, Accessed 18th May 2022.
164
165
- 166 HWE.DESIGN (2020). Hardware engineering design. <https://www.hwe.design/theories-concepts/system-response-stability>. Accessed 8th June 2022.
167
- 168 IBM (2004). What is Class Diagram? <https://developer.ibm.com/articles/the-class-diagram/>, Accessed 18th May 2022.
169
- 170 Interaction Design (2017). User interface (ui) design patterns. <https://www.interaction-design.org/literature/topics/ui-design-patterns>. Accessed 9th June 2022.
171
- 172 javarevisited (2021). Top 10 microservices design patterns and principles - examples. <https://javarevisited.blogspot.com/2021/09/microservices-design-patterns-principles.html>. Accessed 9th June 2022.
173
174
- 175 Kaspersky (2021). What are bots? – definition and explanation? <https://www.kaspersky.com/resource-center/definitions/what-are-bots>. Accessed 9th June 2022.
176
- 177 Kasunic, M. and Anderson, W. (2004). Measuring systems interoperability: Challenges and opportunities. Technical Report CMU/SEI-2004-TN-003, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA. <http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=6871>. Accessed 1st June 2022.
178
179
180
- 181 Kruchten, P. (1995). The 4+1 View Model of architecture. *IEEE Software*, 12(6):42–50. <https://doi.org/10.1109/52.469759>, Accessed 18th May 2022.
182
- 183 MuleSoft (2015). What is an api? (application programming interface). <https://www.mulesoft.com/resources/api/what-is-an-api>. Accessed 8th June 2022.
184
- 185 nginx (2021). What is load balancing? <https://www.nginx.com/resources/glossary/load-balancing/>

186 d-balancing/. Accessed 9th June 2022.

187 Overby, S., Greiner, L., and Paul, L. G. (2017). What is an sla? best practices for service-level
 188 agreement. [https://www.cio.com/article/274740/outsourcing-sla-definitions-and](https://www.cio.com/article/274740/outsourcing-sla-definitions-and-solutions.html)
 189 [-solutions.html](https://www.cio.com/article/274740/outsourcing-sla-definitions-and-solutions.html). Accessed 11th June 2022.

190 Paycore (2019). Service levels. [https://paycore.io/slas/payment-and-payout-gateway](https://paycore.io/slas/payment-and-payout-gateway-sla/)
 191 [-sla/](https://paycore.io/slas/payment-and-payout-gateway-sla/). Accessed 10th June 2022.

192 PayPal (2016). Paypal payments pro. [https://www.paypal.com/us/webapps/mpp/paypal-p](https://www.paypal.com/us/webapps/mpp/paypal-payments-pro)
 193 [ayments-pro](https://www.paypal.com/us/webapps/mpp/paypal-payments-pro). Accessed 9th June 2022.

194 Richardson, C. (2020). Pattern: Api gateway / backends for frontends. [https://microser](https://microservices.io/about.html)
 195 [vices.io/about.html](https://microservices.io/about.html). Accessed 9th June 2022.

196 Robinson, M. (2019). What does federated login mean? a simple, detailed an-
 197 swer. [https://carvesystems.com/news/what-does-federated-login-mean-a-simple-](https://carvesystems.com/news/what-does-federated-login-mean-a-simple-detailed-answer/)
 198 [detailed-answer/](https://carvesystems.com/news/what-does-federated-login-mean-a-simple-detailed-answer/). Accessed 1st June 2022.

199 Schwarzer, V. (2022). Lecture 11 – risk storming and how to be a good architect [powerpoint
 200 slides]. Software Architecture 506 SWA, Worms University of Applied Sciences.

201 Smartsheet (2017). Download free, customizable risk matrix templates. [https://www.sm](https://www.smartsheet.com/all-risk-assessment-matrix-templates-you-need)
 202 [artsheet.com/all-risk-assessment-matrix-templates-you-need](https://www.smartsheet.com/all-risk-assessment-matrix-templates-you-need). Accessed 10th June
 203 2022.

204 SPD LOAD (2019). How much does it cost to develop an app in 2022? cost break-
 205 down. [https://carvesystems.com/news/what-does-federated-login-mean-a-simpl](https://carvesystems.com/news/what-does-federated-login-mean-a-simple-detailed-answer/)
 206 [e-detailed-answer/](https://carvesystems.com/news/what-does-federated-login-mean-a-simple-detailed-answer/). Accessed 1st June 2022.

207 Steel, C., Nagappan, R., and Lai, R. (2012). *Core Security Patterns: Best Practices and*
 208 *Strategies For J2EE, Web Services, and Identity Management*. Pearson PTR, Boston.

209 techopedia (2018). Wrapper. [https://www.techopedia.com/definition/4389/wrapper-s](https://www.techopedia.com/definition/4389/wrapper-software-engineering)
 210 [oftware-engineering](https://www.techopedia.com/definition/4389/wrapper-software-engineering). Accessed 9th June 2022.

211 UN (2022). Stop food loss and waste, for the people, for the planet. *un.org*. [https://www.un](https://www.un.org/en/observances/end-food-waste-day)
 212 [.org/en/observances/end-food-waste-day](https://www.un.org/en/observances/end-food-waste-day), Accessed 18th May 2022.

213 Vilmate (2019). Payment gateway integration in a mobile application. [https://vilmate.co](https://vilmate.com/blog/payment-gateway-integration-in-a-mobile-application/)
 214 [m/blog/payment-gateway-integration-in-a-mobile-application/](https://vilmate.com/blog/payment-gateway-integration-in-a-mobile-application/), Accessed 31st may
 215 2022.

216 Waykar, Y. (2015). role of use case diagram in software development. *International Journal*
 217 *of Management and Economics*. [https://www.researchgate.net/publication/3229918](https://www.researchgate.net/publication/322991847_role_of_use_case_diagram_in_software_development)
 218 [47_role_of_use_case_diagram_in_software_development](https://www.researchgate.net/publication/322991847_role_of_use_case_diagram_in_software_development). Accessed 1st June 2022.

219 Wikipedia (2020). Security pattern. https://en.wikipedia.org/wiki/Security_pattern.
 220 Accessed 9th June 2022.

221 Zoho (2019). Online payment gateway – definition, working & benefits. [https://www.zoho](https://www.zoho.com/books/guides/payment-gateways.html)
 222 [.com/books/guides/payment-gateways.html](https://www.zoho.com/books/guides/payment-gateways.html). Accessed 9th June 2022.

223 **Abbreviations**

224 **API** Application Programming Interface.

225 **DoS** Denial of Service.

226 **FAO** Food and Agriculture Organization of the United Nations.

227 **SLA** Service Level Agreement.

228 **UN** United Nations.

229 Glossary

230 **Application Programming Interface (API)** Software intermediary that promotes the com-
231 munication between different systems/applications/ softwares [MuleSoft (2015)].

232 **Denial of Service (DoS)** Intentional interruption or laming of network services.

233 **Service Level Agreement (SLA)** Contract between a service provider and a consumer re-
234 garding the expected level of service that is accepted. This contract is based on defined
235 metrics, measures to remediate occurring problems and penalties if the contracted level
236 is not achieved [Overby et al. (2017)] .

237 **Activity Diagram** This kind of diagram shows the behavior of a system, it depicts in a graph-
238 ical fashion the logic of a single use case [Baresi (2009)].

239 **API Gateway** Server used an single entry point into a system. It forwards requests to the used
240 service. It is broadly used for authentication, auditing and logging services [Richardson
241 (2020)].

242 **App** It refers to the mobile application to be developed.

243 **bots** Short for robot. It is a software that performs automated and repetitive tasks. It is
244 usually controlled by a malicious actor who targets a network or service. They can
245 be used consume resources and make a service unavailable, steal credentials and other
246 attacks [Kaspersky (2021)].

247 **Class Diagram** This kind of diagram presents the structure of a system with its classes,
248 attributes, methods and relationships [IBM (2004)].

249 **Client** Since we have two major stakeholders that will use the app, the word client will specify
250 the one that places an order in the app.

251 **Federated Login** Authentication method in which users use existing accounts to gain access
252 to another domains or systems without the need of creating new credentials. The
253 authenticity of a user is attested by service and granted to another [Robinson (2019)].

254 **Intercepting Validator** Mechanism used to check if user's input (data or file) corresponds to

255 the criteria defined in the app. Non-conform input are discarded before even reaching
256 the app [Kaspersky (2021)].

257 **Load Balancer** Device used to distribute traffic/resource/request across different servers, so
258 one of them are not overloaded [nginx (2021)].

259 **Microservice** Software architecture approach made of small independent services used to
260 communicate with other resources like APIs. The advantage of using this architecture
261 is its scalability and maintainability, since each service is responsible for a very small
262 group of correlated tasks [AWS (2017)].

263 **Mobile Payment Gateway** Those services works as an intermediary between customer, mer-
264 chant and bank/credit card company. Here a payment request is sent to the gateway
265 and forwarded to the approval instances. The core functionality of those gateways is
266 the cryptography within the communication steps [Vilmate (2019)].

267 **Provider** The second major stakeholders are those who offer their products. They can be
268 restaurants, bakeries, pastries and similar.

269 **Risk Assessment** Report used to identify risk/weakness that may exist in a project. [Schwarzer
270 (2022)].

271 **Stakeholder** Describes all kind of potential person or entity that may have interest using the
272 app.

273 **System Response** The output of a system after an input [HWE.DESIGN (2020)]. [Robinson
274 (2019)].

275 **Use Case Diagram** This kind of diagram presents the main requirements and functionali-
276 tiesPa of a systems. It displays a simplified overview of core purpose of the application
277 [Waykar (2015)].

278 **User** See stakeholder.

279 **Wrapper** Element used to encapsulate the complexity of one entity so it can be processed by
280 another entity [techopedia (2018)].