
Software Requirements Specification

for

Pixel Pizza

Version 2.4

Prepared by

Parv Desai: 17IT434

Darshan Tripathi: 17IT442

Acknowledgement: Sections of this document are based upon the IEEE Guide to Software Requirements Specification (IEEE 830)

Table of Contents

Table of Contents	ii
Revision History	iii
1. Introduction	1
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Product Scope	2
1.5 References	2
2. Overall Description	3
2.1 Product Perspective	3
2.2 Product Functions	3
2.3 User Classes and Characteristics	4
2.4 Operating Environment	5
2.5 Design and Implementation Constraints	5
2.6 User Documentation	6
2.7 Assumptions and Dependencies	6
3. External Interface Requirements	7
3.1 User Interfaces	7
3.2 Hardware Interfaces	7
3.2 Software Interfaces	8
3.3 Communications Interfaces	8
4. System Features	9
4.1 Order Ready-made Pizza	9
4.2 Order Custom Pizza	10
4.3 Orders Management	11
4.4 Account Management	12
5. Other Nonfunctional Requirements	13
5.1 Performance Requirements	13

5.2	Safety Requirements	13
5.3	Security Requirements	13
5.4	Software Quality Attributes	13
Appendix A: Glossary		14
Appendix B: Analysis Models		15
Appendix C: List of References		18

Revision History

Name	Date	Reason for Changes	Version
Parv Desai	20-1-2020	Initial version	1.0
Darshan Tripathi	05-2-2020	Minor Changes	1.1
Darshan Tripathi	07-2-2020	Added new content	2.0
Parv Desai	10-02-2020	Added Diagrams	2.1
Darshan Tripathi	14-02-2020	Added new functionalities and diagrams	2.3
Parv Desai	16-02-2020	Minor Changes	2.4

1. Introduction

1.1 Purpose

The new Pixel Pizza app takes the experience of ordering pizza to a whole new level. Users will be able to select each and every component of their pizza with this app. New customers will also be able to order from the store's bestselling and best combinations. Users will be able to order pizza for online delivery or takeaway via app.

1.1.1 Vision Statement

The aim of this app is to enable consumers to order the pizza of their own likings and taste. Also provide restaurant to get insight about the likings and disliking of the mass

1.1.2 Scope

The scope of this app includes development of an interactive user interface to provide step by step creation of their pizza. Provide the restaurant a simpler way to manage the orders.

1.2 Document Conventions

RN	React Native
DB	Database

Table 1: Document conventions

1.3 Intended Audience and Reading Suggestions

This document is to be read by the development team, the project managers, marketing staff, testers and documentation writers. Our stakeholders, company manufacturing associated hardware, company providing embedded operating system, shareholders of Pixel Pizza and distributors who markets the finished product, may review the document to learn about the project and to understand the requirements. The SRS has been organized approximately in order of increasing specificity. The developers and project managers need to become intimately familiar with the SRS. Others involved need to review the document as such:

Overall Description – Marketing staff have to become accustomed to the various product features in order to effectively advertise the product.

System features – Testers need an understanding of the system features to develop meaningful test cases and give useful feedback to the developers.

External Interface Requirements –The hardware developers need to know the requirements of the device they need to build. The marketing staff also needs to understand the external interface requirements to sell the product by describing the user-friendly features of the Pixel Pizza.

1.4 Product Scope

The Pixel Pizza has features that enable consumers to order the in-house made or custom pizza, they can also choose whether they want a takeaway or home delivery. Customers can pay online or have cash on delivery service. Restaurants will also have a simpler way to manage orders.

1.5 References

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=720574&isnumber=15571>

2. Overall Description

2.1 Product Perspective

Overcoming the downsides of current apps in market which do not provide the ordering of custom-made pizza this application is developed by the app developing team at Pixel Pizza which will enable the customers of Pixel Pizza to make the pizza of their choice and order them from home. The App being developed is for portable smartphones for both platforms i.e. iOS and android. The current version is version 1.0.0 and a whole new product for our organization. The product will work in coordination with the floor working of the Pixel Pizza outlet. The restaurant will also be able to add or remove or modify the products they sell.

2.2 Product Functions

Home screen: Welcome users on a page which shows the offers and some pre made pizzas which they can order directly.

Custom Pizza Maker: The users will be prompted to select base, sauce and toppings of their choices from the list. The price of base will be according to the diameter of pizza. The price of sauce and toppings will be constant.

Checkout: The customers will be able to review their selection, select the size of pizza and specify the number of pieces to be cut the pizza in and pay for the order and checkout.

Profile: The user will be able to view his addresses, edit the addresses and edit his profile overall.

History: The user will be able to view his previous orders.

Orders Management: The restaurant will be able to see the pending and served orders from their side.

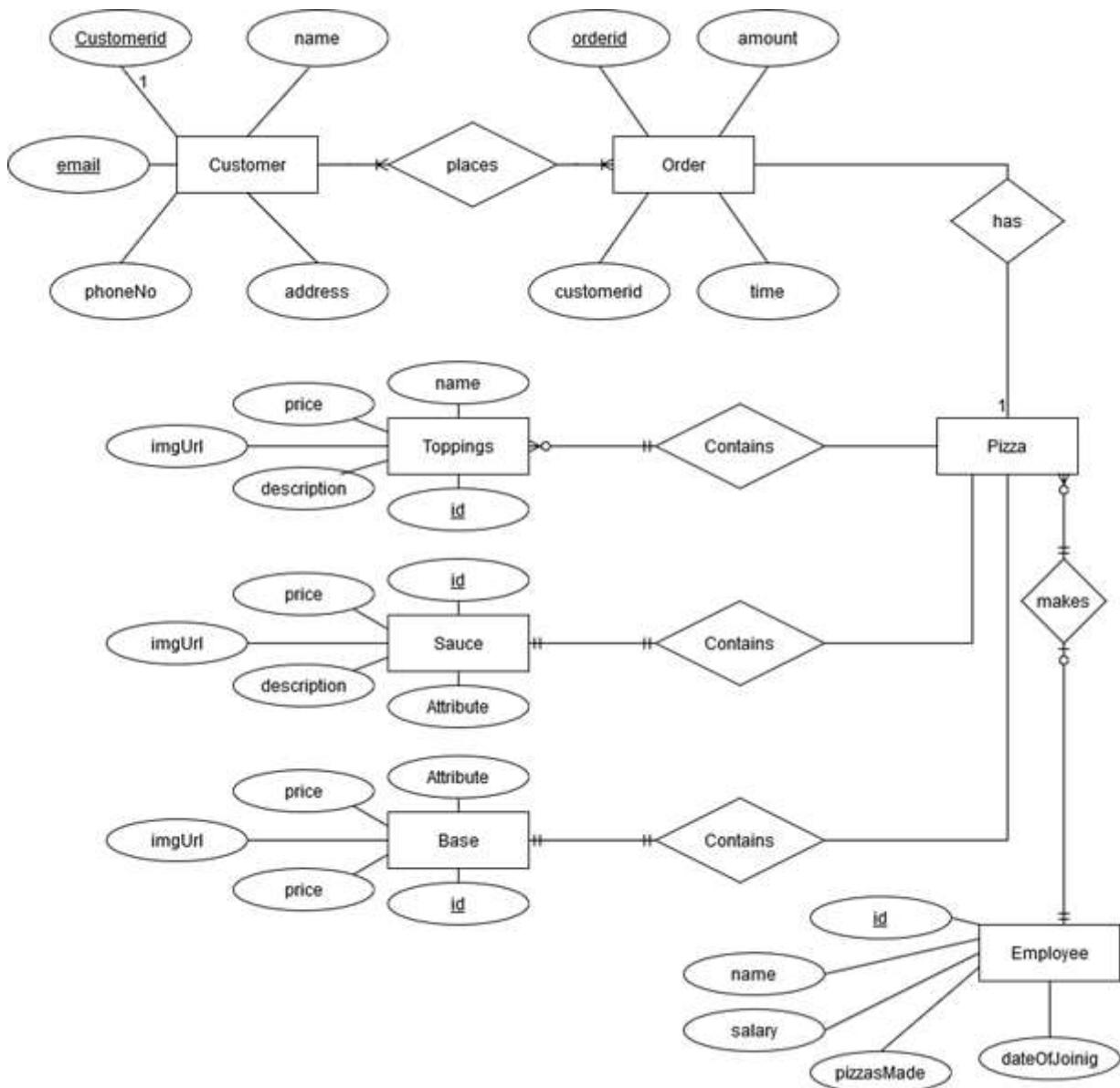


Figure 1: ER Diagram

2.3 User Classes and Characteristics

Customer: The ones who will use the app to order the pizza through the app. Pay online/ on delivery and get catered

- Sign up
- Login
- Make Pizza

- Edit profile
- Review Order
- Place order
- Make payment

Manager: The person in-charge of the items in stock, their price. He will perform his functions through a web portal. He will be the only person to make changes in the database.

- Add item
- Remove item
- Modify Item

Employee: The persons working in store who will make the pizza and deliver it. They will be able to see the pending orders and change the status of it according to the progress of production of pizza.

- Accept Order
- Reject Order

2.4 Operating Environment

The app will be able to run on any iOS smartphone and all smartphones with Android version 4.0+.

The database will be hosted on cloud provided service

- Centralized database
- Client/Server System
- Operating system: Android/iOS
- Database: MySQL
- Platform: React-native, JavaScript, PHP

2.5 Design and Implementation Constraints

The developers are constrained by the time frame available for development. This time constraint will lead the developers to have less time for debugging and UI features. Also, the app should be

lightweight so that it can run different architecture of devices. The less time frame will also lead to sacrificing some UI/UX on the vendor side.

- Use of Centralized database
- Multiple Components increases redundancy
- SQL queries for such schema

2.6 User Documentation

No user manual, online tutorial is made available with it. But the customer will be guided through the app through the very simple design of the app and very simple UI and minimal features.

2.7 Assumptions and Dependencies

The knowledge of the user is made that he knows English because the app is available in only one language. The user has sound knowledge about how the online food ordering works. The user also has an e-wallet to pay online. The workers are also assumed to have a basic knowledge of how databases work.

3. External Interface Requirements

3.1 User Interfaces

The theme for this app is black and orange thus few uses of colors will make the app look minimal which most of the users like. Also, very few uses of functional components in the screen at a time will lead to less confusion to customers while using the app.

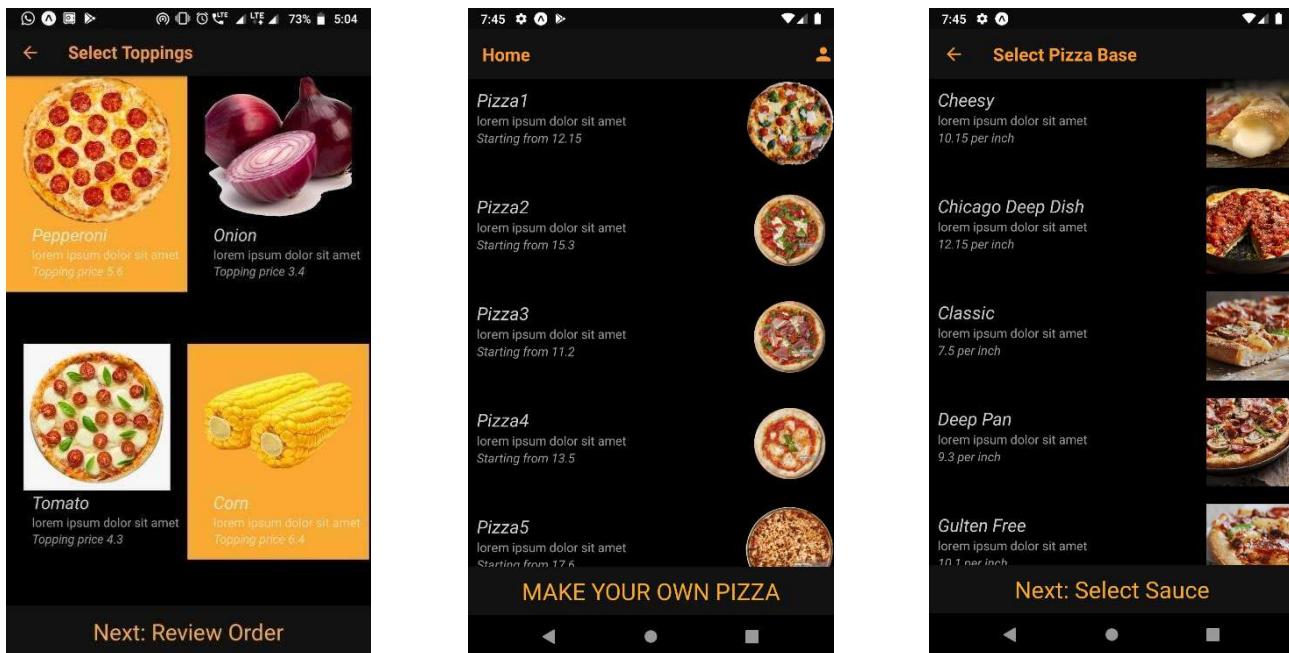


Figure 2: Screenshots

3.2 Hardware Interfaces

- A server to serve API calls
- Device which supports React-native builds

3.3 Software Interfaces

The data will be fetched in each screen from the database via API calls. The data will be converted into json format in the server and will be served to the app. The app will convert into human readable format and will display on the screen.

Also, when user logins in our app, the user_Id is stored in the local storage of mobile itself and hence whenever the user opens the app again, it opens in logged in state if he/she hadn't logged out previously.

The details of the user are sent to the server, where it is stored in a database using PHP code, when he/she registers.

Operating system	Android/iOS
Database	MySQL
Back-end	PHP
Front-end	React-native
External Libraries	react-native-elements, AsyncStorage, react-navigation, react-navigation-stack

Table 2: Software Interfaces

3.4 Communications Interfaces

The communication between the different parts of the system is important since they depend on each other. However, in what way the communication is achieved is not important for the system and is therefore handled by the underlying operating systems for app. Our app will use API calls to communicate with Server.

- Users will be notified of their order status via emails/SMS
- Employees will have an automatic reloading page to see the orders

4. System Features

4.1 Order Ready-Made Pizza

4.1.1 Description and Priority

A user who is registered to our app and is logged in, can order a pizza from available options, which is to be delivered to a specific address given by him. He/She may cancel or change the order if it has not yet been prepared, but it will lay an extra charge as fine to the user by the restaurant.
PRIORITY=HIGH

4.1.2 Stimulus/Response Sequences

- Stimulus: User requests to place an order for one or more pizzas from available options.
- Response: App queries User for details of pizza(s), payment and delivery address.
- Stimulus: User requests to cancel order.
- Response: If status is “Accepted,” App allows users to cancel order with some fine charges.

4.1.3 Functional Requirements

Place Order: The app shall let a User who is logged into the Pixel Pizza App place an order for one or more pizza meals.

Order Pay Method: When a user is placing orders, the system shall ask the user to select a payment method.

4.2 Order a Custom Pizza

4.2.1 Description and Priority

The user on the home screen will be prompted to order his custom pizza as per his/her choices. The user will select a crust of his choice, then automatically be headed to a list of sauce offerings and select from it. Then after selecting the sauce of his choice he can choose a range of toppings. And finally, he can select size and the number of cuts he wants to make in his pizza.

PRIORITY=MEDIUM

4.2.2 Stimulus/Response Sequences

Stimulus: The user requests to make custom pizza

Response: The app fetches the data from database and shows him the available options

Stimulus: User makes the choice from the given list

Response: The app will store the choice he made in local storage and will forward to next screen

4.3.3 Functional Requirements

Pizza preparation: The app will require several screens to display the option of different components of pizza from which the user will select an option.

Persistent storage: The app will store the user's choice into the local storage. So that users may not lose their choice.

Payment: Finally, the users can place order by selecting size and cuts they want to make.

4.3 Orders Management

4.3.1 Description and priority

The store on their side will be able to see new incoming orders and mark them as completed, making, or on the way according to the progress. The restaurants will have full control on the incoming orders and their preparation process.

PRIORITY=LOW

4.3.2 Stimulus/Response Sequences

Stimulus: New order comes in the database

Response: The orders page on the store side will be updated

Stimulus: The order status changed by employee

Response: The database will be changed and the change will be reflected in the store side page and the user will also be notified.

4.3.3 Functional Requirements

Set status of order: The employee of the store can change the status of the order as completed, making or on the way.

See the orders list: The employee can see the outstanding orders to be completed with the different components which the user has requested. Along with the list of completed orders.

4.4 Account Management

4.4.1 Description and priority

A login screen will be displayed whenever a user installs the app, and he/she gives his/her details to System registers or logsins itself. All the data of the user can be seen by the Restaurant. User can view the history of orders and also edit his/her profile.

PRIORITY=HIGH

4.4.2 Stimulus/Response Sequences

Stimulus: The user registers itself to the app for the first time.

Response: The app now allows user to login in app.

Stimulus: The user logins into the app with his/her email and password.

Response: The app allows users to access further functionality of the app.

Stimulus: The user requests to view the order history and all details of previous orders.

Response: The app displays the list of previous orders with all details.

Stimulus: The user requests to edit his/her profile.

Response: The app allows you to change any user information.

4.4.3 Functional Requirements

Login/Registration: The app will require 2 screens to provide the functionality of Login and Registration to the user.

Persistent storage: The app will store the user_Id to local storage so that it will always be logged in whenever he/she opens the app without logging out.

View History: The app will fetch the transaction history using API calls and display it.

Edit Profile: The app will edit any information of the user and updates it in database.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Load time of the user interface shall not take longer than 10 seconds to load onto screen after the user opens the app. The login information shall be verified within 5 seconds of submitting the information. At Least 50-100 users can use our app during the peak hours without any delay or overloading from the app side. The database performance is made quick and correct by the INNODb engine provided by the MySQL database. The back end will have the performance guarantee of that of the Apache run servers.

5.2 Safety Requirements

If there is extensive damage to a wide portion of the database/server due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to in another database of another server and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

5.3 Software Quality Attributes

- Availability: The App should be available to maximum users possible especially during peak hours according to the working of the server.
- Maintainability: The User should be able to maintain the history of all orders he/she had ordered.
- Correctness: The User should be given correct information about the available pizza(s).
- Portability: The User can access his/her account on any device by logging into his/her account on another device.
- Usability: The App is designed for both Android and iOS. Also, the language used is English which is standard all over the world.

Appendix A: Glossary

App: This abbreviation is used for the term Application used in mobile.

API: Application Programming Interface (API) is an intermediary software that allows 2 applications to communicate.

MySQL: It is an open-source relational database management system.

UI/UX: This abbreviation is used for User Interface Design which means the design of any machines and software, such as computers, home appliances, mobile devices, and other electronic devices, with the focus on maximizing usability and the user experience.

PHP: It is a popular general-purpose scripting language that is especially suited to web development.

Appendix B: Analysis Models

Data Flow Diagram:

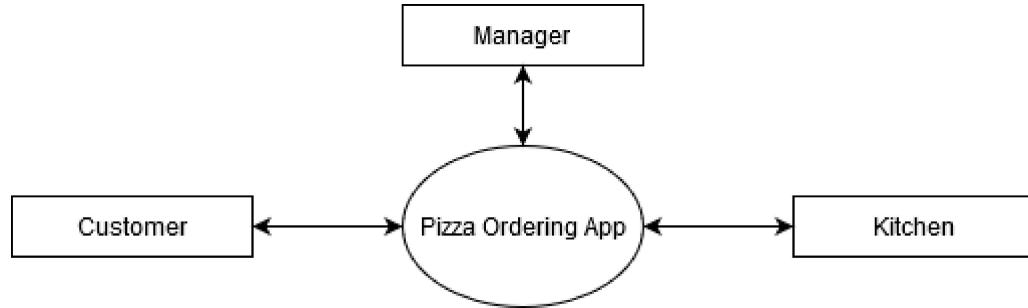


Figure 3: Context Level Data Flow Diagram

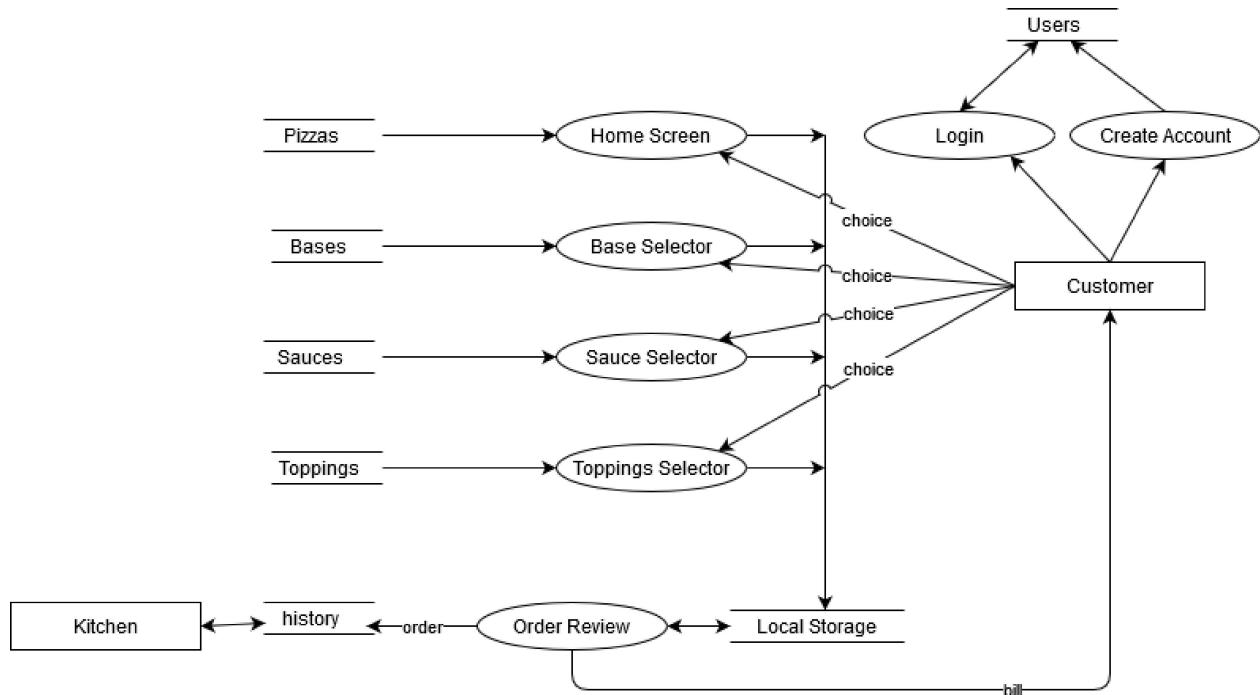


Figure 4: Level 1 Data Flow Diagram

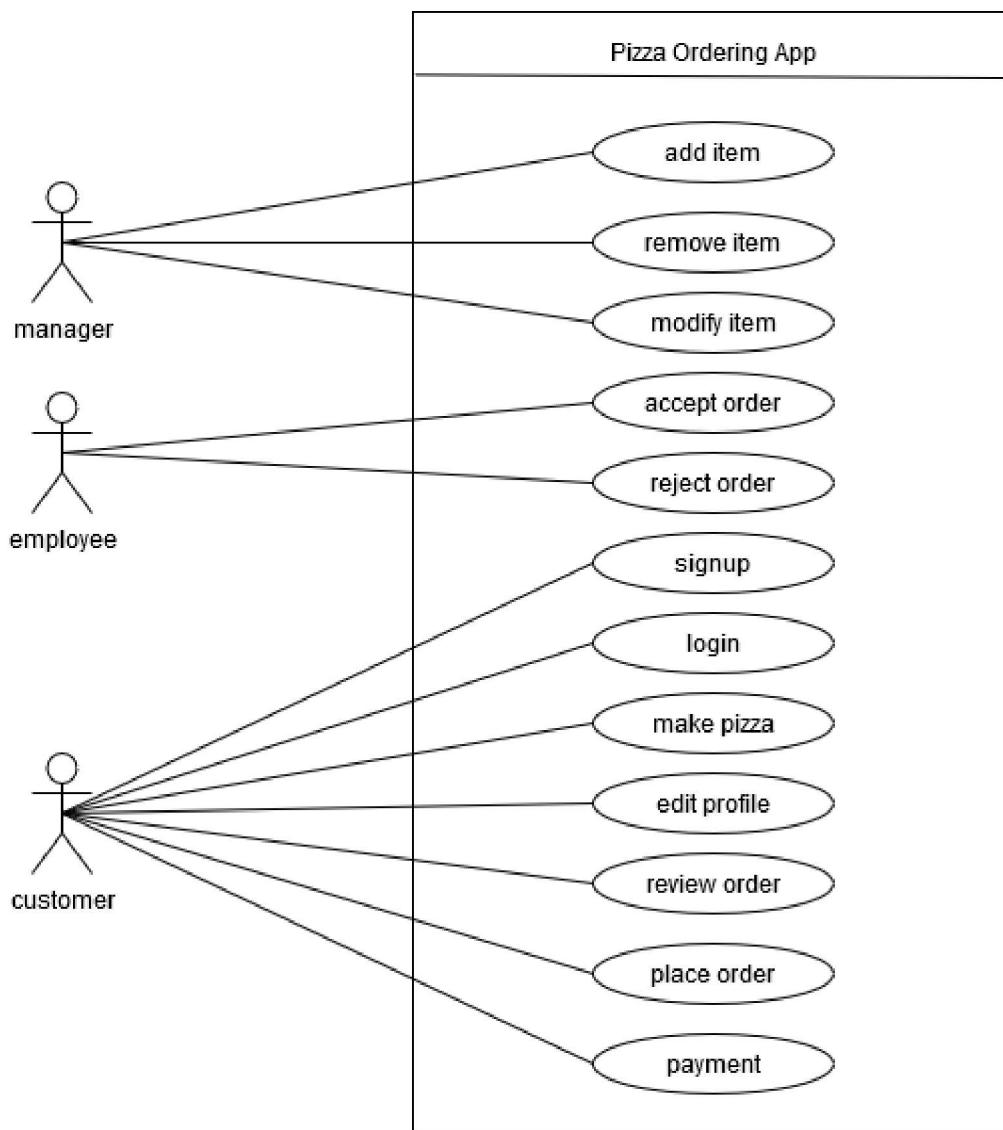
Use Case Diagram:

Figure 5: Use Case Diagram

Class Diagram:

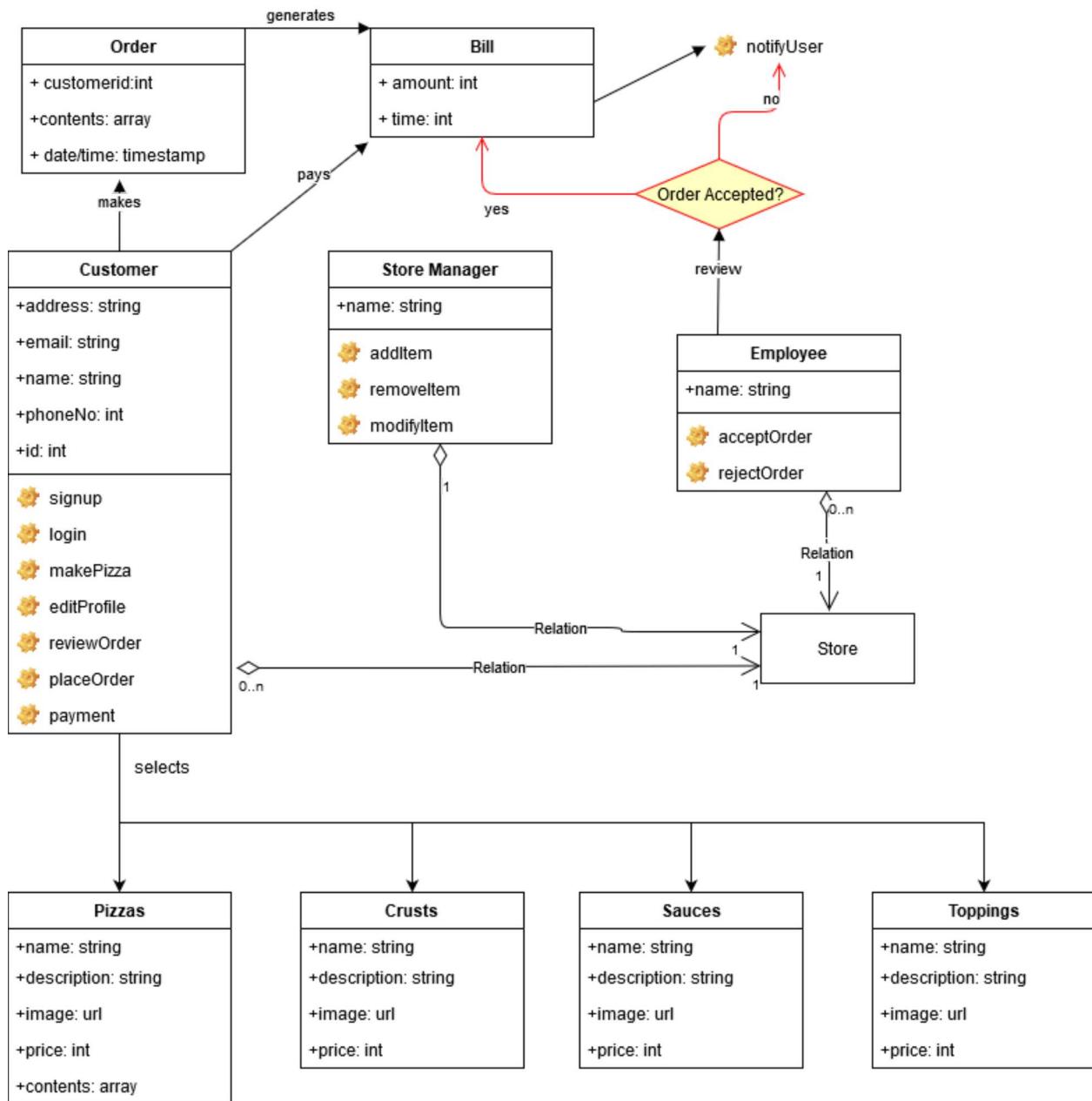


Figure 6: Class Diagram

Sequence Diagram:

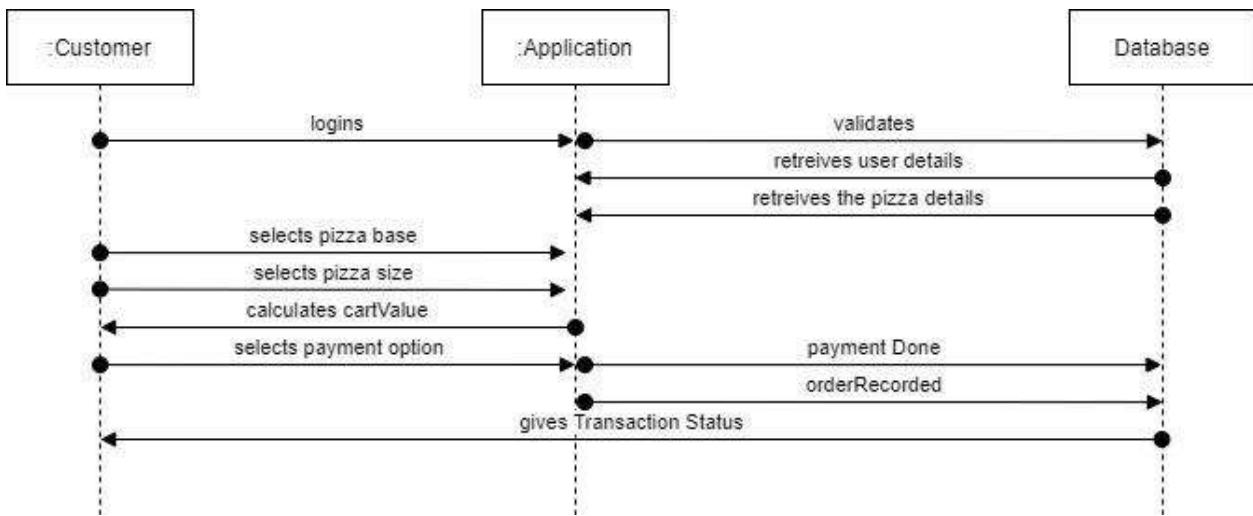


Figure 7: Sequence Diagram

Appendix C: List of References

1. IEEE Recommended Practice for Software Requirements Specifications," IEEE Std 830-1998
2. Bhargave, Ashutosh, Niranjan Jadhav, Apurva Joshi, Prachi Oke, and S. R. Lahane. "Digital ordering system for restaurants using Android." International journal of scientific and research publications 3, no. 4 (2013): 1-7.