TED UNIVERSITY

Department of Computer Engineering

2022 Fall

CMPE 491 Senior Project I

High-Level Design Report

Name of the Project: Receipt Analyzer

The URL of webpage: https://bzelihagunay.wixsite.com/receipt-analyzer

Ahmet Berat Akdoğan, Başar Aslan, Burçak Zeliha Günay, Hüseyin Hikmet Fındık

Supervisor: Aslı Gençtav

Jury Members: Gökçe Nur Yılmaz, Venera Adanova

Date of Submission: 03/01/2023

Table of Contents

1.Introduction

Today, with the opening of new supermarkets or groceries, an increase in the product range is observed. However, due to the changing inflation rate in our country, the prices of various products change frequently. Considering that a person constantly goes to the market to meet their needs in a daily life, it becomes difficult to observe the changes in the prices of the products, with frequent price changes. Even though some people check their receipts after grocery shopping, some people just throw them in the trash without even looking at their receipts.

People who keep their receipts can barely observe how much the price of a specific product they want has changed in the long term by saving or keeping these receipts in a certain part of their home or by taking pictures of them. However, these methods are not quite efficient. Because, as a customer, we sometimes want to look for a price related to a specific product, and this price tracking process becomes complicated. Therefore, it may not be enough to keep the past receipts at home. In addition, sometimes the stored receipts can be lost from the phone or from the house.

Nowadays, although some mobile applications have the features of scanning images, a mobile application created based on the market products of the scanned data is not a quite common idea. In addition, in our project, we plan to transform the information about that product to a visual graphic after the market product receipts are scanned. The implementation of this idea distinguishes our project from other software in the industry. Thus, users will be able to observe the product they want much better by looking at the time/ price graphic. In this context, our project differs from other ideas in the sector.

In this project, we thus want to design a mobile application that will make it easier to observe the price changes of products by using OCR technology. OCR technology basically stands for “Optical Character Recognition” which converts images into text and gets the desired parts from the text. With the help of our mobile application, customers can scan the receipt they purchased from the market by OCR technology and observe the price change of products they desire.

1.1 Purpose of the System

Receipt Analyzer is a mobile application that enables customers to understand the price changes of products purchased from supermarkets. The main purpose of the app is to allow customers to scan the receipt that includes a list of products. These contain some necessary information such as product name, price of product, and date. After scanning the receipt, Receipt Analyzer plots a time/price graph which compares the scanned price of products and price of products that already exist in the database in a specific period.

Some factors played a major role in the use of a new idea in our project. In the country we live in, due to inflation reasons, the prices of products are constantly changing, regardless of the sector. In this way, we want to provide an awareness about the prices of the products that change frequently, while providing convenience for the users to observe the changing price rates. Furthermore, it is aimed to provide users with the opportunity to effectively observe the price changes in products in a certain period with a time/price graph.

1.2 Design Goals

Receipt Analyzer enables its users to compare the prices of products in the most effective and easy way. For this reason, we have various design goals in our project to provide the best experience to users. In order to provide the best experience with a working and stable system, there are some items that we decide to include directly or indirectly in the design. These are listed as follows.

1.2.1 Extensibility

In our project, our goal is to create a software system with major capabilities such as usage of Optical Character Recognition to scan the receipt, Time/Price Graph to illustrate price change of products, and a welcoming page that includes log-in and register buttons, etc. Therefore, by considering these major points in the project, we are aiming to provide an extensible system without changing the basic architecture or major parts of our software. For example, if the circumstances meet, we can add additional features to our project such as adding new supermarkets to the app, increasing product variety or options for login and register, or adding a language option to provide a more customizable UI and increase extensibility.

1.2.2 Maintainability

In our project, we aim to design components of the application which are integrated to each other. Therefore, in case of any change, update, or action we take during the process, we aim to integrate the action without affecting the current components in the system. For instance, we are planning to scan receipts from one marketplace in the project. If there is a need for adding another market to our program which means a new receipt structure in the scanning procedure, we aim for users to continue the current market scanning process without a negative effect to provide maintainability.

1.2.3 Usability

The User Interface (UI), appearance, and menu of the application are aimed to be prepared simple and understandable, as our application can be used from elders to young people. So that the application is not complicated, we will not overwhelm the screen with as many buttons and texts as possible. The fact that the application can be used on mobile in order to provide a comfortable and easy service to the user greatly increases the usability rather than providing on the website. People will be able to read the receipt they keep at home without any bother, by just clicking a single button on their mobile.

1.2.4 Response Time and Performance

Since the rate of use of smartphones by people is quite high, we aim to provide a fast and comfortable experience to users by making the project as a mobile application in order to increase performance. We also plan to minimize the delay in switching to a different screen after the pressed the buttons. Finally, when reading the products, the graphic time to be plotted should not take too long. When it is considered the variety of products in a market, transferring all this information to the database brings a lot of data to be kept. In order to sketch the graph, it may be necessary to query among the tables with very detailed data in the background. In this case, we aim to compare product prices and then graph them in an acceptable time to provide an effective service experience to the customer.

1.2.5 Availability

As developers, we should provide a service that offers the receipt scanning mobile application idea based on the possibility that users can use the application whenever they want. When a problem arises regarding the application in the future, we should make improvements without affecting customers' experience. Being able to offer the service to customers when they need them makes the mobile application available. Therefore, the idea of availability is very essential for the mobile application that is developed.

1.2.6 Scalability

In our project, the availability of products in the database for customers to compare them with their receipts is an important issue for a better experience while using the application. Therefore, to increase the comparison of product experience of customers and meet their expectations about specific products, we aim to keep as many varieties and number of products as possible. In addition, despite the increase in the number of markets, some users do not have the desired markets where they live, instead we come across other market brands around the cities. Although our current goal in the project is to decide on only one market to implement the idea of reading receipts, it is possible to increase the number of market brands that the application supports, with increasing demands and opening new markets.

1.2.7 Security

Information that is written on the receipt such as market details, location, and time of purchase, will be stored only on a database that is permitted to other users. By this way, as a developer, we aim to ensure the security and privacy of users. In addition, the password of the people logging in will be encrypted with the encryption algorithm rather than stay as a plain text and stored in the database as an encrypted version with a confidential and secure manner. Therefore, self-information about users will not be shared with other sources or third parties.

1.2.8 User Interface

User enters the mobile application by clicking the app icon on the smartphone. After the application's interface is opened, an informative screen will welcome users. After that, the register and login buttons will be presented to the user on the main screen. Then, the user who logins the application by clicking the login button, encounters a camera scan button to scan the receipt. As developers, we will ask questions like do you accept the camera permission as we encounter in most of the social media applications, and users will be able to proceed to the next step by pressing the yes or no buttons that we give. By adding a green tick next to the yes button and a red cross next to the no button, we will emphasize that the decision made is important in the user interface. Even if we present only one market to the user, we will add a button there to let the user know what the name of that market is, and we make them ask that market to press the button manually. After pressing the market button, we will provide users with an effective button that is centered on the phone screen to open the camera. After pressing this button, the UI seems like a QR code reading screen and allows us to scan the receipt. In the application interface, as a last step, the user encounters a time/price graph which is scaled on the mobile screen to visualize price changes of products. We may use different colors for the current data on the database and scanned data to illustrate the price change of products.

1.2.9 Portability and Compatibility

Since the receipt scanning feature can be implemented more easily and efficiently on a mobile phone camera rather than a computer camera, we think that this idea will work more efficiently on mobile applications rather than on the website, so the application will be presented on a mobile system. Therefore, it is aimed to perform the project in an Android environment, which means that users should have a smartphone running on Android Operating System whose version is not too old.

1.3 Definitions, Acronyms, and Abbreviations

- **Mobile Application:** A mobile application or app is a computer program or software application designed to run on a mobile device such as phone, tablet, or watch.
- **Receipt Analyzer:** The name of our project which is planned as an app to scan market receipts.
- **OCR:** Optical Character Recognition is a process that converts an image of text into a machine-readable text format.
- **Database:** A database is an organized collection of structured information, or data, typically stored electronically in a computer system. Considering the wide variety of products in a market, for this project, it is important to store detailed information about the products in the database after the receipts are scanned.
- **UI:** The point of human-computer interaction and communication in a device. Since the idea of scanning receipts provides its service on the mobile application, we should be considering the user interface carefully as developers.
- **Website:** A website is a collection of web pages and related content that is identified by a common domain name and published on at least one web server.
- **Sketch/Plot:** Sketch/Plot is a graphical technique for representing a data set, usually as a graph showing the relationship between two or more variables. In our project, we use this technique to graphically display the long-term price and name information of the scanned products to the users.
- **Developer:** A developer is an individual that builds and creates software and applications. He or she writes, debugs, and executes the source code of a software application.
- **Encrypted:** It is a converted data from a readable, plaintext format into an unreadable, encoded format: ciphertext. We are planning to use this methodology to hide the password of customers who register the application.
- **Plaintext:** In cryptography, plaintext is usually ordinary readable text before it is encrypted into ciphertext, or readable text after it is decrypted.
- **QR Code:** A quick response (QR) code is a type of barcode that can be read easily by a digital device and which stores information as a series of pixels in a square-shaped grid. In this context, QR, which is a kind of scanning and barcode method, is like the OCR technology that is implemented in our project of scanning market receipts.
- **Android (Operating System):** A Linux-based mobile operating system that primarily runs on smartphones and tablets.

1.4 Overview

Receipt Analyzer is a mobile application, most probably released in Android platform, targeted for mobile phones. Main idea of the application is to demonstrate price changes of a product in a specific grocery for its customers. Users who use the application will have the opportunity to observe the difference in the price change of the product with the help of a visual material such as a time/price graphic. In the application, users can scan the market receipt thanks to their mobile phones to start the price analyzing process. The system will first read and process the market receipt completely. Then, it will recognize the necessary information (product name, date, and price) on the receipt and transfer it to the system. The database will serve to hold the necessary information detected at this point. As a last step, users can be able to see the time/price graph of a specific product in the application to observe the differences between the products already written on the database and scanned products more effectively. In conclusion, once the project is completed successfully, we allow customers to compare product prices between the past and the present using their smartphone.

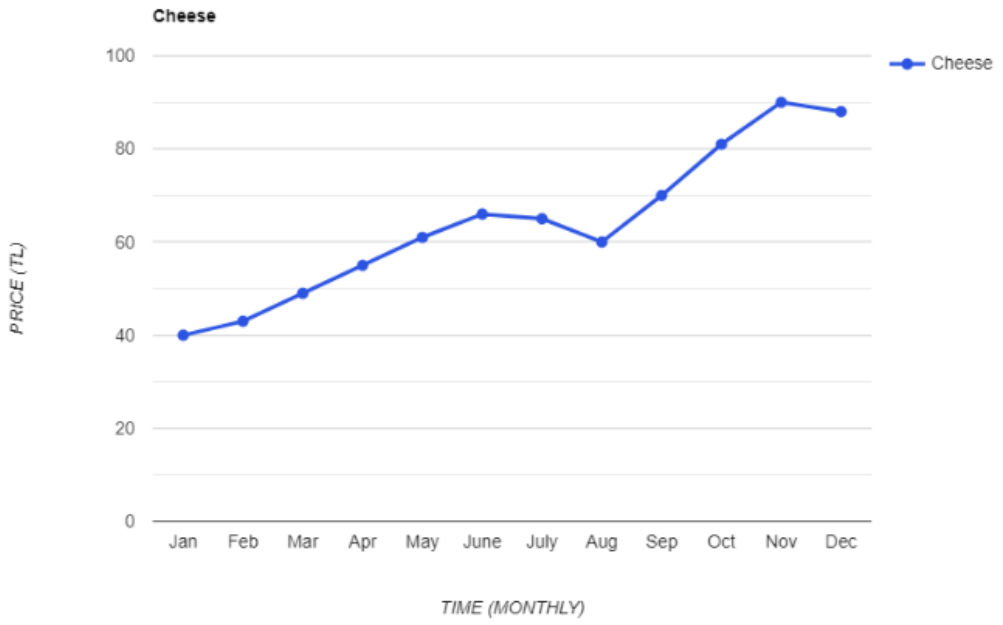

Figure-1: Time/Price Graph of Cheese Product.

Customers can observe price change of products on certain months as the above figure. After the user scans the receipt, user can compare the scanned product price and products that are already exist in the database on the graph.

The point where the graph starts is when the product is first added and saved to the database. So, in that moment, price can be shown as a reference point. After this process, the prices of the products whose receipts are scanned may increase or decrease according to this reference point.

2. Current Software Architecture (if any)

Since the goal in this project is to create awareness in the grocery sector about price change of products in a specific period, it is aimed to combine our idea with some software concepts and tools to implement the project. Image processing technology, which is an area of Computer Science, is used to read market receipts. More specifically, OCR (Optical Character Recognition) technology is used when developing the project. Furthermore, there are some methods that we have not decided, but we will continue with one of them. The first option is to use library solutions in Python such as PyTorch, OpenCV or Tesseract OCR engine. Alternatively, we consider Firebase's ML Kit and Text Recognition solution. We are planning to use Realtime Database, a solution of Firebase, in order to save the required scanned data (product name, date, product price) in the database. After all, retrieve it from the database and draw the graph. Alternatively, SQLite, NoSQL, which is compatible with Python, can be used in database selection. After the image processing period of the project, since the project will be developed on a mobile application on Android in our project, we plan to use one of the alternatives such as Java, Kotlin, or Dart (Flutter) as a programming language.

Any further changes related to the project about the image processing or programming language that we are going to use will be stated accordingly with the new possible decisions from the group members.

3 Proposed Software Architecture

3.1 Overview

The proposed software architecture includes the design of the mobile application, which will scan the market receipts and record the name, price, and date of purchase of the product purchased from the market by image processing in a database. The proposed software architecture includes the design of all systems and data structures necessary for the application to run.

3.2 Subsystem Decomposition

Subsystems, which are defined as separate modules that can perform different functions of the application, are an important building block that facilitates the functionality and use of the application. Therefore, the correct decomposition of subsystems will positively affect the performance and usability of the application.

Subsystems can be created by grouping them according to the functionality of the application. For example, one subsystem can access the database while another subsystem can create the user interface. Also, subsystems can be thought of as separate processes that perform different parts of the application. For example, one subsystem might add data to the application's database, while another subsystem might read data from the database.

The decomposition of subsystems is carried out during the design phase of the application. This process aims to determine which subsystems will perform the functions of the application. In this way, the structure of the application is better understood, and the development of the application becomes easier.

For example, the subsystems specified in 3.2.1 are planned for the mobile application that scans the market receipts and records the name, price, and date of purchase of the product purchased from the market by image processing in a database:

3.2.1 Sample Subsystem Decomposition Recommendations

Database management subsystem: This subsystem handles access to the application's database. This subsystem performs operations such as adding data to the database, reading data, updating data, and deleting data. For example, this subsystem allows the products taken from the scanned market receipts to be recorded in the database. This subsystem can use methods such as REST API or direct SQL queries to access the database.

Image processing subsystem: This subsystem extracts the product name, price, and date of purchase from the scanned market receipts. This subsystem can be implemented using an image processing library such as OpenCV.

User interface subsystem: This subsystem creates the user interface of the application. This subsystem performs an interface design where the user can list the products in the database and show the change in the price of the products over time. This subsystem can be implemented using a mobile application development framework such as Flutter.

These subsystems are the parts that are foreseen to be used together to perform the functions of the application.

3.2.2 Advantages of Sample Subsystem Decomposition Recommendations

Database management subsystem: Thanks to the use of REST API, database access operations can be performed quickly and easily from within the application. In addition, thanks to the use of direct SQL queries, more comprehensive queries can be performed in accordance with the structure of the database. In this way, the desired information from the database can be retrieved more quickly and easily.

Image processing subsystem: With the use of an image processing library such as OpenCV, the process of extracting the product name, price and purchase date from the scanned market receipts can be performed quickly and effectively. In this way, while the information extracted from the scanned market receipts is recorded in the database, erroneous or incomplete information is prevented.

User interface subsystem: With the use of a mobile application development framework such as Flutter, user interface design can be accomplished quickly and easily. Also, thanks to the use of Flutter, the app can be run on both Android and iOS devices. In this way, the usability of the application can be made available to a wide range of users.

Thanks to the combined use of these subsystems, the functions of the application can be performed more effectively.

3.3 Hardware/Software Mapping

Since it is designed as a mobile application, choosing a suitable operating system and programming language for mobile devices will be important in hardware/software mapping. For example, it can be developed in Java or Kotlin for Android devices and Swift for iOS devices. It is planned to use Flutter or React Native for a cross-platform build. An appropriate management tool will be selected for database management, and data will be stored and accessed. For example, options such as Firebase Realtime Database or SQLite can be considered. By using the REST API, the application's database access operations will be performed. Thus, market receipts scanned by the user can be added to the database and price tracking can be done at any time.

3.4 Persistent Data Management

Persistent data management ensures that the changes made to the database in our application are stored permanently, even in cases such as closing the application or resetting the device. Therefore, SQLite can be used as a database in our application. SQLite is a small and lightweight database and can be used easily on mobile devices. This database will be used to store the information of the market receipts scanned in our application. In addition, SQL queries will be used for accessing this database. In this way, our application will be able to quickly access and update the information of the market receipts scanned by the user.

Another alternative is to provide persistent data management for our mobile application using Firebase Realtime Database. Firebase Realtime Database is a data storage service provided by Google and has a real-time database feature. With this feature, data can be shared between the users of your application and the data can be updated instantly. Also, Firebase Realtime Database offers an easily usable and configurable API for mobile apps.

Therefore, the use of Firebase Realtime Database can be evaluated for our mobile application that scans market receipts.

3.5 Access Control and Security

- Access rights to the database in our application will be given only to authorized users. In this way, only authorized users can access the database and the information in the database can be accessed.

- Authorized users will be identified by performing a specified login process within the application. Security measures such as username and password can be applied for this login process.
- Access rights to the database can be determined according to the roles of the users within the application. For example, administrative users may have full access rights to the database, while normal users may only have access to certain information in the database.
- The security of the information in the database is also important. Therefore, the information in the database can be stored encrypted. In this way, the security of the information in the database is ensured.
- In addition, security protocols such as SSL (Secure Sockets Layer) can be used during the process of accessing information in the database. These protocols ensure that database access operations are performed securely.
- Performing certain operations by users with database access rights during database access operations.
- For example, operations such as updating product information in the database can only be performed by administrative users. In this way, the accuracy and up-to-datedness of the information in the database is ensured.
- In addition, timeout periods can be determined for database access operations. For example, during database access operations, the number of transactions a user can make in an hour can be limited. In this way, attempts to perform a large amount of access to the database are prevented.
- IP address restrictions may also be applied in database access operations. For example, only certain IP addresses may be allowed to access the database. In this way, it can be applied to the database as a measure to ensure security.
- It is also important that the passwords used in database access operations are changed frequently. In this way, the risk of learning passwords over time is reduced.
- It is also important that users' access rights to the database can be revoked in database access procedures. For example, when a user's access rights to the database are revoked, that user no longer has access to the database. In this way, security measures are taken to the database.

3.6 Global Software Control

For the proposed software architecture, a software control framework such as Redux can also be used for global software control of the application. Such frameworks help in managing the state and data of the application and make it easy to monitor changes that occur during the application's run. Alternatively, a structure such as Flux can be used for global software control of the application. Such structures also help in managing the data flow of the application and make it easier to monitor the changes that occur during the application's operation.

3.7 Boundary Conditions

For the proposed software architecture, the boundary conditions of the application should also be considered. For example, there may be no internet connection while the application is running, and in this case, database access operations may not be performed. Therefore, a local data store should also be created where the data can be stored in cases where the application has no internet connection. In this way, the changes that occur while the application is running can be saved in this local data store and this data can be transferred to the database when the internet connection is restored.

3.7.1 Initialization

For the proposed software architecture, the actions to be taken during application initialization should be considered. For example, operations such as establishing the database connection required for the application to work and loading the data in the database must be performed. In addition, the settings and previously entered data necessary for the use of the application must be loaded. In this way, the application can be ready for use.

3.7.2 Termination

For the proposed software architecture, the actions to be taken during the termination of the application should also be considered. For example, operations such as transferring the data from the local data store created during the operation of the application to the database should be done. In addition, the release of resources used during the operation of the application should also be done. In this way, errors that may occur during the termination of the application are prevented.

3.7.3 Failure

For the proposed software architecture, the errors that may occur during the execution of the application should also be considered. For example, database access operations may fail while the application is running, and how the application will behave in this case must be determined. For the proposed software architecture, an error management mechanism should be established that can detect errors while the application is running to prevent such errors.

4 Subsystem Services

Receipt Analyzer, an application that will scan receipts and record data of product names, product prices, and dates of purchase, will have various subsystem services. Those subsystems are different predicted tools to comply with the planned functions of the app and we aim to ensure working quality of functions of the app through common use of these subsystem services.

The app relies on several subsystem services in order to function properly. These subsystems handle specific tasks that are essential to the app's main function of scanning receipts and recording prices.

4.1 Receipt Scanning Service

This service is responsible for taking an image of a receipt and using optical character recognition (OCR) technology to extract the text from it. The service first converts the image into a digital format and then applies OCR algorithms to identify and extract the text. The service also performs pre-processing on the image to improve the accuracy of the OCR process, such as removing noise and correcting perspective distortion. The extracted text is then passed on to the next subsystem for processing.

4.2 Text Processing Service

This service takes the text extracted from the receipt and processes it to extract the relevant information such as the names and prices of individual products. The service uses natural language processing (NLP) techniques to analyze the text and identify patterns and keywords that indicate the presence of product names and prices. The service then extracts these pieces of information and stores them in a structured format for further processing.

4.3 Price Comparison Service

This service is responsible for comparing the prices of products over time. It does this by storing the prices of products in a database and using this data to generate graphs and other visualizations that allow the user to see how the prices have changed. The service also provides features such as price alerts, which allow the user to set up notifications to be sent when the price of a certain product reaches a certain threshold.

4.4 Notification Service

This service is responsible for sending notifications to the user when the prices of certain products reach a certain threshold. The service integrates with the Price Comparison Service to receive updates on price changes and sends notifications to the user via email, push notification, or in-app message.

4.5 Security Service

This service is responsible for ensuring the security of the app and its data. It handles tasks such as user authentication, data encryption, and access control. The service uses secure protocols and technologies such as HTTPS and AES to protect the data transmitted between the app and the server, and it implements robust user authentication and access control mechanisms to prevent unauthorized access to the app and its data.

Overall, these subsystem services work together to provide a seamless experience for the user and ensure that the app is able to accurately scan receipts and track the prices of products over time.

5. Glossary

Mobile Application	A mobile application or app is a computer program or software application designed to run on a mobile device such as phone, tablet, or watch.
Receipt Analyzer	The name of our project which is planned as an app to scan market receipts.
OCR	Optical Character Recognition is a process that converts an image of text into a machine-readable text format.
Database	A database is an organized collection of structured information, or data, typically stored electronically in a computer system. Considering the wide variety of products in a market, for this project, it is important to store detailed information about the products in the database after the receipts are scanned.
UI	The point of human-computer interaction and communication in a device. Since the idea of scanning receipts provides its service on the mobile application, we should be considering the user interface carefully as developers.
Website	A website is a collection of web pages and related content that is identified by a common domain name and published on at least one web server.
Sketch/Plot	Sketch/Plot is a graphical technique for representing a data set, usually as a graph showing the relationship between two or more variables. In our project, we use this technique to graphically display the long-term price and name information of the scanned products to the users.
Developer	A developer is an individual that builds and creates software and applications. He or she writes, debugs, and executes the source code of a software application.
Encrypted	It is a converted data from a readable, plaintext format into an unreadable, encoded format: ciphertext. We are planning to use this methodology to hide the password of customers who register the application.
Plaintext	In cryptography, plaintext is usually ordinary readable text before it is encrypted into ciphertext, or readable text after it is decrypted.
QR Code	A quick response (QR) code is a type of barcode that can be read easily by a digital device and which stores information as a series of pixels in a square-shaped grid. In this context, QR, which is a kind of scanning and barcode method, is like the OCR technology that is implemented in our project of scanning market receipts.
Android (Operating System)	A Linux-based mobile operating system that primarily runs on smartphones and tablets.

6. References

- *Mobile app. (n.d.). In Wikipedia. Retrieved January 3, 2023, from https://en.wikipedia.org/wiki/Mobile_app*

- *What is a database? (n.d.). In Oracle. Retrieved January 3, 2023, from https://www.oracle.com/database/what-is-database/*

- *Website. (n.d.). In Wikipedia. Retrieved January 3, 2023, from https://en.wikipedia.org/wiki/Website*

- *Developer. (n.d.). In Techopedia. Retrieved January 3, 2023, from https://www.techopedia.com/definition/17095/developer*

- *Data encryption: Definition and related FAQs. (n.d.). In Druva. Retrieved January 3, 2023, from https://www.druva.com/glossary/what-is-data-encryption-definition-and-related-faqs/*

- *Plaintext. (n.d.). In TechTarget. Retrieved January 3, 2023, from https://www.techtarget.com/searchsecurity/definition/plaintext*

- *Plot (graphics). (n.d.). In Wikipedia. Retrieved January 3, 2023, from https://en.wikipedia.org/wiki/Plot_(graphics)*

- *Quick response (QR) code. (n.d.). In Investopedia. Retrieved January 3, 2023, from https://www.investopedia.com/terms/q/quick-response-qr-code.asp*

- *Software extensibility: Definition, attributes, and techniques. (n.d.). In PeerSpot. Retrieved January 3, 2023, from https://www.peerspot.com/articles/software-extensibility-definition-attributes-and-techniques*

- *Line graph. (n.d.). In RapidTables. Retrieved January 3, 2023, from https://www.rapidtables.com/tools/line-graph.html*