Software Requirements Specification

for

Cryptocurrency Analysis WebApp

Version 1.0 approved

Prepared by Anushraya Sharma

KIET Group Of Institutions

07 April 2023

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The cryptocurrency Analysis WebApp is a web-based application which aims to solve this problem - designed to present all information in a structured and centralized manner. Ranging from history and analysis of past performance to reliable predictions about future dips and rises, this covers them all. Adding on top of these, handy tools like real time value, local currency conversions, monitor of top performing currencies, and other performance statistics are also integrated.

1.2 Document Conventions

Font and Formatting: Use a consistent font and formatting throughout the document to ensure readability and consistency. For example, use a sans-serif font like Arial or Helvetica, and use bold or italic text sparingly to emphasize important points.
Section Headings: Use clear and descriptive headings for each section of the SRS to help readers quickly find the information they need. For example, use "Introduction" for the opening section, "Functional Requirements" for the section describing the system's features, and "Non-functional Requirements" for the section describing performance and quality attributes.
Requirement Numbering: Use a consistent numbering scheme for each requirement statement to facilitate traceability and management. For example, use a hierarchical numbering scheme like "1.1.2" to indicate the section, subsection, and requirement number.
Priority and Importance: Use a clear and consistent priority scheme to indicate the relative importance of each requirement. For example, use "High", "Medium", and "Low" to indicate the urgency and criticality of each requirement.
Dependencies and Relationships: Clearly state any dependencies or relationships between requirements to help readers understand the overall structure and impact of the system. For example, indicate when one requirement is dependent on another, or when a set of requirements are related to a specific feature or function.
Glossary and Definitions: Include a glossary of terms and definitions to clarify any technical or domain-specific terms used in the document. For example, include definitions for terms like "cryptocurrency", "local currency conversions", and "real time value" to ensure a common understanding among all stakeholders.

1.3 Intended Audience and Reading Suggestions

Intended Audience:
Developers: This document will be primarily used by the development team to understand the functional and non-functional requirements of the Cryptocurrency Analysis WebApp.
Project Managers: This document will be used by the project managers to plan and monitor the project's progress and ensure that the requirements are met.
Testers: This document will be used by the testing team to develop test cases and ensure that the system meets the specified requirements.
Documentation Writers: This document will be used by the documentation team to create user manuals and other instructional materials.
Reading Suggestions:
For an overview of the system's purpose and scope, start with the Introduction section.
For a detailed description of the system's functional requirements, refer to the Functional Requirements section.
For a description of the system's non-functional requirements, refer to the Non-functional Requirements section.

For a description of the system's design and architecture, refer to the Design section.
For a description of the system's testing approach and methodology, refer to the Testing section.
For a list of known issues and limitations, refer to the Bugs section.
It is recommended that readers start with the overview sections and then proceed to the sections that are most pertinent to their role or interest. Developers should focus on the functional and non-functional requirements, while testers should focus on the testing section. Project managers should review the entire document to ensure that the project is on track and that all requirements are met.

1.4 Product Scope

The crypto based web-app has a wide scope in the upcoming world as the awareness regarding the cryptocurrency investments is increasing day by day therefore people are more lean towards these cryptos and want to make better investments in these, this webapp can find applications in the stock markets and crypto transactions as well as can be used by various industries and tech giants to make profit and boost the growth of their company and the economy by making a fruitful and profitable investments.

1.5 References

1.Androulaki, E., Karame, G. O., Roeschlin, M., Scherer, T., & Capkun, S. (2013). Evaluating user privacy in bitcoin. In Financial Cryptography and Data Security (pp. 34-51). Springer Berlin Heidelberg.
2.Barber, S., Boyen, X., Shi, E., & Uzun, E. (2012). Bitter to better—how to make bitcoin a better currency. In Financial Cryptography and Data Security (pp. 399-414). Springer Berlin Heidelberg.
3.Brito, J. & Castillo, A. (2013). Bitcoin: a Primer for Policymakers. Mercatus Center, George Mason University.
4.Bearman, J. (2015, May). The Untold Story of Silk Road, Pt. 1. Retrieved from Wired.com Website: https://www.wired.com/2015/04/silk-road-1/
5.Bitcoin: A New Global Economy. (2015, August 4). Retrieved July 2016, from BitPay, Inc. Website: https://blog.bitpay.com/bitcoin-a-new-global-economy/
6.Bovaird, C. (2016, June 24). Bitcoin Rollercoaster Rides Brexit As Ether Price Holds Amid DAO Debacle. Retrieved June 2016, from CoinDesk Website: http://www.coindesk.com/bitcoin-brexitether-price-rollercoaster/

2. Overall Description

2.1 Product Perspective

Cryptocurrencies form a important aspect and part of our economy, and as it is a fairly new emerging domain, the information resources are widely scattered. This proves to be a major hurdle for potential and new investors as well who want to learn about this domain. Combining information from multiple resources - evaluating which ones to trust at the same time, and doing all this at regular intervals of time (considering the volatility of cryptos) increases the slope of the learning curve involved even more.

2.2 Product Functions

Our web-app aims to solve this problem - designed to present all information in a structured and centralized manner. Ranging from history and analysis of past performance to reliable predictions about future dips and rises, this covers them all. Adding on top of these, handy tools like real time value, local currency conversions, monitor of top performing currencies, and other performance statistics are also integrated.

2.3 Characteristics

The cryptocurrency is the trend of today's generation and people are investing in these cryptocurrencies to gain maximum benefit from these investments hence a centralized data is required so that people can analyze the various cryptocurrencies and can make the best possible investment based on the market price, trends, market cap of these cryptos hence a crypto tracker web app came to rescue as they provide a centralized system where people can watch and compare cryptos before making any investments hence to secure and maximize the profit.

2.4 Operating Environment

Operating Environment:
Hardware Platform: The Cryptocurrency Analysis WebApp will operate on any device with a modern web browser and an internet connection, including desktop computers, laptops, tablets, and mobile phones. The minimum required specifications for the hardware platform are a modern processor and at least 4GB of RAM.
Operating System: The Cryptocurrency Analysis WebApp will be compatible with the major operating system, including Windows. The minimum required operating system version is Windows 7.
Web Browser: The Cryptocurrency Analysis WebApp will be accessible via any modern web browser, including Google Chrome, Mozilla Firefox, Apple Safari, and Microsoft Edge. The minimum required browser version is the latest stable release.
Other Software Components: The Cryptocurrency Analysis WebApp will depend on several open-source software components, including React.js, Material UI, CSS, JavaScript, and Chart.js. These components will be integrated into the web application to provide the necessary functionality and user interface.
It is important to note that the operating environment described above is subject to change based on the needs of the project and the evolving requirements of the web application. It is recommended that the development team regularly review and update the operating environment to ensure that the software remains compatible with the latest technologies and platforms.

2.5 Design and Implementation Constraints

Technology Stack: The development team has been instructed to use React.js, Material UI, CSS, JavaScript, and Chart.js for the development of the Cryptocurrency Analysis WebApp. No other front-end frameworks or libraries are allowed.
Security: The web application must follow industry-standard security practices, including encryption of user data, protection against SQL injection attacks, and secure storage of user passwords.
Performance: The web application must be responsive and provide real-time data updates. The development team must optimize the code and minimize the number of requests to the server to ensure smooth performance.
Browser Compatibility: The web application must be compatible with all modern web browsers, including Google Chrome, Mozilla Firefox, Apple Safari, and Microsoft Edge.

User Interface Design: The web application must follow a consistent user interface design and adhere to Material UI guidelines. The design must be visually appealing and easy to use.
Database: The web application will use a MySQL database for storing user data and transaction history.
Maintenance: The development team will be responsible for maintaining the web application after deployment, including bug fixes, security updates, and feature enhancements.
It is important to note that the design and implementation constraints described above may change over the course of the project as new requirements and limitations are identified. The development team should regularly review and update the constraints to ensure that the web application remains compliant with the latest industry standards and best practices.

2.6 User Documentation

On-line Help: An on-line help system will be available within the web application to provide users with quick access to information on specific features and functionality.
Tutorials: A set of tutorials will be provided to help users get started with the web application, including step-by-step instructions on how to create an account, how to add funds, and how to perform basic transactions.
Delivery Format: All user documentation will be delivered in digital format, accessible through the web application itself or downloadable in PDF format.
It is important to note that the user documentation components described above may change over the course of the project as new requirements and needs are identified. The development team should regularly review and update the user documentation to ensure that it remains up-to-date and helpful for users.

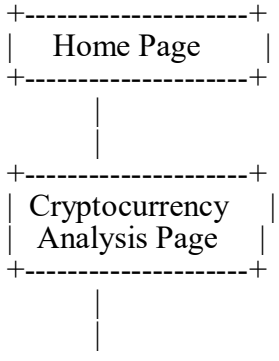
2.7 Assumptions and Dependencies

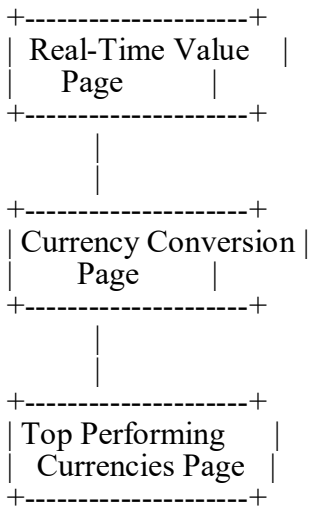
Assumptions:
The availability and accuracy of cryptocurrency market data feeds: The application's ability to provide reliable and up-to-date information on cryptocurrency prices and trends relies on the availability and accuracy of data feeds from third-party providers.
The availability of external APIs: The application may rely on external APIs to perform certain functions, such as currency conversions. Assumption is that these APIs will be available and reliable.
The availability of the necessary infrastructure: The application will require a stable and secure infrastructure, including servers, storage, and networking equipment. Assumption is that these resources will be available and sufficient to support the application's needs.
Dependencies:
ReactJS, MaterialUI, ChartJS and other third-party libraries: The application relies on various third-party libraries and components for its user interface, data visualization, and other features.
Web browser compatibility: The application is designed to work on modern web browsers such as Google Chrome, Mozilla Firefox, and Microsoft Edge, and may not function properly on older or less common web browsers.
Internet connectivity: The application requires a stable and reliable internet connection for users to access and use the application.
Security considerations: The application must adhere to security best practices and may rely on various security technologies and services, such as encryption and authentication, to protect user data and transactions. These dependencies may change over the course of the project as new requirements and technologies are identified.

3. External Interface Requirements

3.1 User Interfaces

Home Page: The home page displays an overview of the top cryptocurrencies, their current prices, and trends.
Cryptocurrency Analysis Page: This page includes detailed analysis of selected cryptocurrencies. It displays charts, graphs, and other statistical data related to the selected cryptocurrency.
Real-Time Value Page: This page displays the real-time value of various cryptocurrencies. It includes a search bar to search for a specific cryptocurrency.
Currency Conversion Page: This page allows users to convert the value of cryptocurrencies into their local currency.
Top Performing Currencies Page: This page displays the top-performing cryptocurrencies based on market capitalization, price, and other performance metrics.
Here's a diagram illustrating the different components of the User Interface:





3.2 Hardware Interfaces

The cryptocurrency analysis web application is a web-based application that does not have any direct hardware interfaces. It will run on any device with a web browser and an internet connection. However, the application relies on third-party APIs to fetch real-time cryptocurrency data, and the reliability of these APIs can affect the performance of the application. The APIs must be compatible with the application and able to provide reliable data for the application to function as intended.

3.3 Software Interfaces

The software interfaces refer to the connections between the cryptocurrency analysis web application and other software components or applications. These interfaces are essential for the application to work correctly and provide all the necessary functionalities to the end-users.
The cryptocurrency analysis web application will interact with various software components, including databases, operating systems, tools, libraries, and integrated commercial components. The following software interfaces have been identified:
Database Interface: The cryptocurrency analysis web application will store and retrieve data from a database management system. The application will use SQL queries to interact with the database.
Operating System Interface: The application will be designed to run on multiple operating systems, including Windows, Linux, and macOS.
Charting Library Interface: The application will use Chart.js library to generate charts and graphs based on cryptocurrency data.
Currency Conversion API Interface: The application will use an external API to retrieve real-time currency conversion rates for different currencies.
Web Server Interface: The application will communicate with the web server to handle user requests and responses.
Browser Interface: The application will be accessible via web browsers and will utilize various web technologies such as HTML, CSS, and JavaScript to provide the user interface.
Third-Party API Interface: The application may use third-party APIs to retrieve data related to cryptocurrency prices and trends.
The data shared across these software components will include user input, cryptocurrency data, real-time currency conversion rates, and various other system data. The communication between these components will be facilitated using standard protocols and interfaces, and detailed application programming interface (API) protocols will be documented separately.

3.4 Communications Interfaces

The communications interfaces for this product include:
Email: The system should be able to send email notifications to users for various events such as account creation, password reset, and new message alerts. The email format should conform to standard internet email protocols and support HTML formatting.
Web browser: The system will be accessed through a web browser and should be designed to be compatible with major web browsers such as Google Chrome, Mozilla Firefox, and Microsoft Edge.
Network server communications protocols: The system should be able to communicate with network servers using standard protocols such as TCP/IP, HTTP, and HTTPS.
Electronic forms: The system should be able to receive and process electronic forms submitted by users. The forms should conform to standard formats such as PDF, DOC, and DOCX.
Communication security or encryption issues: The system should be designed to use secure communication protocols to ensure the privacy and security of user data. All sensitive data should be encrypted during transmission using standard encryption protocols such as SSL or TLS.
Data transfer rates: The system should be designed to handle high data transfer rates to ensure efficient data transfer between the server and client.
Synchronization mechanisms: The system should be designed to synchronize data between different components using standard synchronization protocols such as SyncML or WebDAV.

4. System Features

4.1 Real-Time Price Tracking

4.1.1 Description and Priority

The web application should be able to track and display real-time prices of various cryptocurrencies, including Bitcoin, Ethereum, Litecoin, and others. Without real-time price tracking, users would have to rely on delayed data which can be misleading and cause them to make uninformed decisions. Therefore, real-time price tracking is a critical feature of any cryptocurrency tracker web application and should be given a high priority.

4.1.2 Stimulus/Response Sequences

User Action: User opens the cryptocurrency tracker web application.

System Response: The web application loads and displays the real-time prices of various cryptocurrencies.

User Action: User selects a specific cryptocurrency to track.

System Response: The web application displays the real-time price of the selected cryptocurrency, and updates it in real-time as the price changes.

User Action: User views the historical data for a specific cryptocurrency.

System Response: The web application displays the historical price data for the selected cryptocurrency, allowing the user to analyze trends and make informed decisions.

4.1.3 Functional Requirements

REQ-1: Data sources: The web application must have access to reliable and up-to-date data sources that provide real-time prices for various cryptocurrencies.

REQ-2: Real-time price updates: The web application must be able to update the price of each cryptocurrency in real-time, without any significant delay.

4.2 Historical Data

4.1.1 Description and Priority

The historical data feature in crypto currency tracker web app allows user to view the past performance of various cryptocurrencies. This feature provides user with the ability to analyze trends and patterns in the market and make informed decisions about buying and selling cryptocurrencies hence has a great priority.

4.1.2 Stimulus/Response Sequences

User Action: User opens the cryptocurrency tracker web application.

System Response: The web application loads and displays the real-time prices of various cryptocurrencies.

User Action: User selects a specific cryptocurrency to track.

System Response: The web application displays the real-time price of the selected cryptocurrency, and updates it in real-time as the price changes.

User Action: User views the historical data for a specific cryptocurrency.

System Response: The web application displays the historical price data for the selected cryptocurrency, allowing the user to analyze trends and make informed decisions.

4.1.3 Functional Requirements

REQ-1: Integration with real-time price tracking: The web application should integrate seamlessly with the real-time price tracking feature, allowing users to switch between real-time and historical data analysis with ease.

REQ-2: Performance Optimization: The web application must be optimized for performance, to ensure that historical data retrieval and analysis occur quickly and smoothly.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Yes, performance requirements are important for a cryptocurrency tracker web application to ensure that it can handle various circumstances and provide users with a smooth and responsive experience. Here are some examples of performance requirements for the app and their rationale:

Response Time: The web application must respond to user actions within a reasonable timeframe, typically within 2-3 seconds. This is important because users expect a fast and responsive experience, especially when viewing real-time price data or making time-sensitive trading decisions.

Scalability: The web application must be scalable to handle increasing user traffic and data volume. This is important because the cryptocurrency market is highly volatile and can experience sudden spikes in activity, which could overwhelm the app's servers and slow down its performance.

5.2 Safety Requirements

Here are some requirements that are concerned with possible loss, damage, or harm that could result from the use of the cryptocurrency tracking web app, along with safeguards and actions that must be taken to mitigate these risks:

Accuracy of data: The web application must ensure the accuracy and reliability of the data it presents to users, to prevent losses resulting from inaccurate or misleading information. This includes verifying data sources and providing clear disclaimers and explanations of any limitations or uncertainties associated with the data.

Legal Compliance: The web application must comply with all relevant laws and regulations related to cryptocurrency trading and financial transactions, to prevent legal liability or harm to users. This includes providing clear and accurate disclosures of any fees or other costs associated with using the app, as well as complying with regulations related to user identification and verification.

5.3 Security Requirements

5.4 Security and privacy are critical concerns for a cryptocurrency tracker web application, as users rely on the app to handle sensitive financial information. Here are some requirements regarding security and privacy issues, along with user identity authentication requirements:

Data Encryption: The web application must use strong encryption methods to protect sensitive data, both in transit and at rest. This includes using SSL/TLS encryption for data transmission and implementing strong encryption algorithms for data storage.

Data Privacy: The web application must protect user privacy by adhering to applicable data privacy regulations and implementing appropriate data privacy measures. This includes obtaining user consent for data collection and processing, providing clear and transparent privacy policies, and implementing privacy-enhancing technologies such as data anonymization or pseudonymization.

5.5 Software Quality Attributes

Usability: The web application should be easy to use, intuitive, and visually appealing. This will help users navigate the app and find the information they need quickly and easily.

Availability: The web application should be available 24/7 to ensure that users can access the app whenever they need it. This will require robust infrastructure and redundancy measures to prevent downtime or service disruptions.

Reliability: The web application should be reliable and consistent in its performance, with minimal downtime or service disruptions. This will require robust testing and monitoring measures to identify and address any performance issues before they affect users.

Maintainability: The web application should be designed for easy maintenance and updates, with clear code organization and documentation, modular design, and automated testing and deployment tools.

5.6 Business Rules

User Roles: The web app may have different user roles such as a regular user, administrator, or moderator, each with specific permissions and restrictions.

Functionality and Features: The app may offer various features and functions, such as real-time price tracking, portfolio management, news and analysis, and trading tools.

User Authentication: Users may be required to create an account and authenticate their identity before accessing the app's functionality.

6. Other Requirements

Internationalization Requirements: The cryptocurrency analysis web application will be designed to support multiple languages and locales. The user interface should be translatable and should support right-to-left languages. The application should also support various date and time formats based on the user's locale.

Legal Requirements: The cryptocurrency analysis web application must comply with all applicable laws and regulations related to data privacy and security. The application must also comply with any laws and regulations related to financial transactions and must ensure that user data is protected and kept confidential.

Reuse Objectives: The cryptocurrency analysis web application should be designed with modularity and reuse in mind. The code should be organized into modules and components that can be easily reused in other projects. The application should also use industry-standard technologies and frameworks to ensure that the code can be easily maintained and updated over time. The application should be open source and should encourage community contributions and feedback.

Appendix A: Glossary

stakeholders: Individuals or groups who have a vested interest in the Cryptocurrency Analysis WebApp, such as developers, project managers, testers, and documentation writers.

Real-time Value: The current value of a cryptocurrency, updated in real-time as the market changes.

Integration: The seamless combination of different features and functionalities within the web application.

Response time: The time it takes for the web application to respond to user actions, typically within 2-3 seconds.

Historical data: The past performance of various cryptocurrencies, used to analyze trends and patterns in the market and make informed decisions about buying and selling.

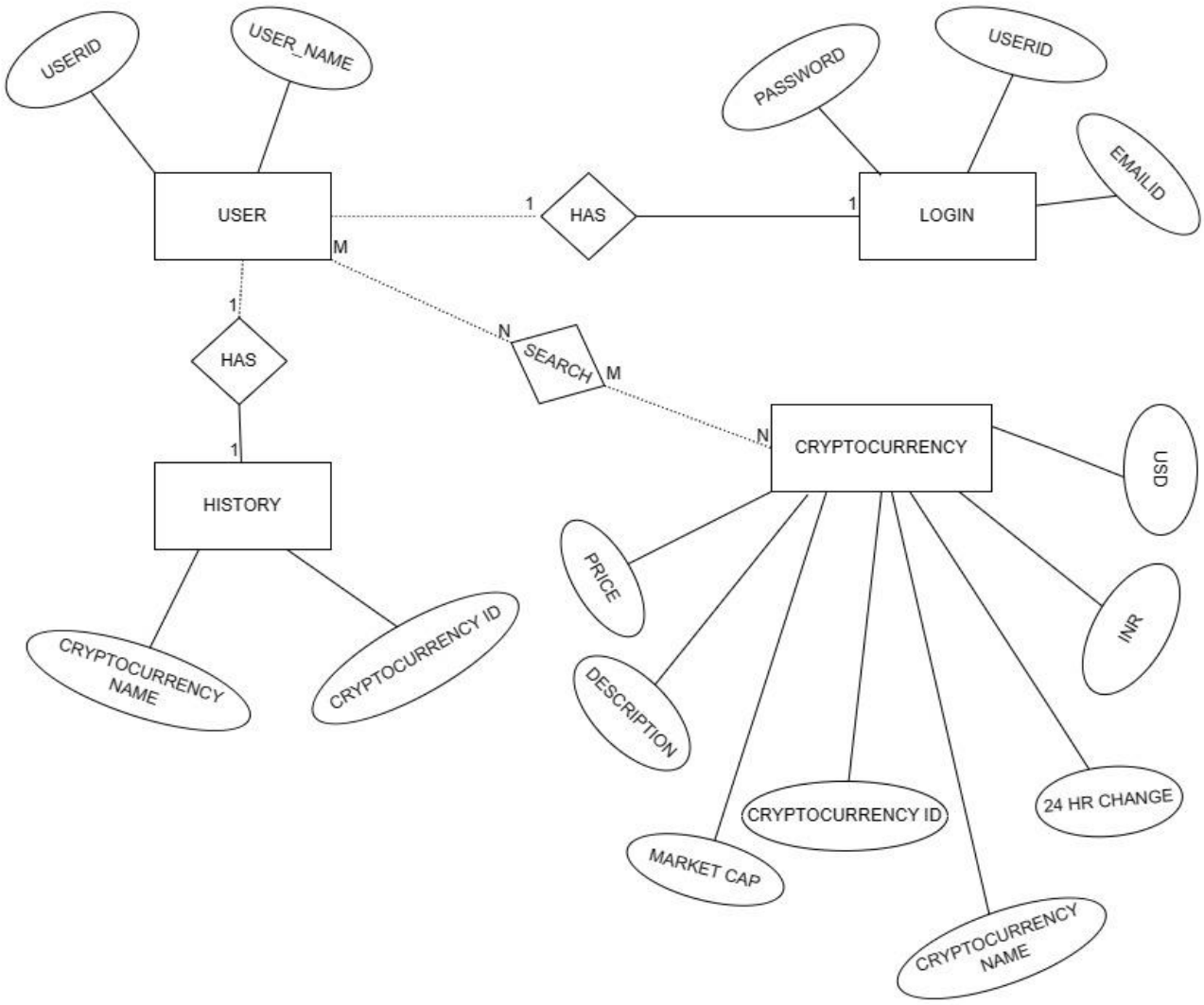
Data feeds: A continuous stream of data provided by a third-party service or application, often used to update or supplement an existing dataset.

API: An Application Programming Interface is a set of protocols and tools for building software applications that define how different software components should interact with each other.

Market capitalization: The total value of a company or asset based on its current market price and the number of outstanding shares.

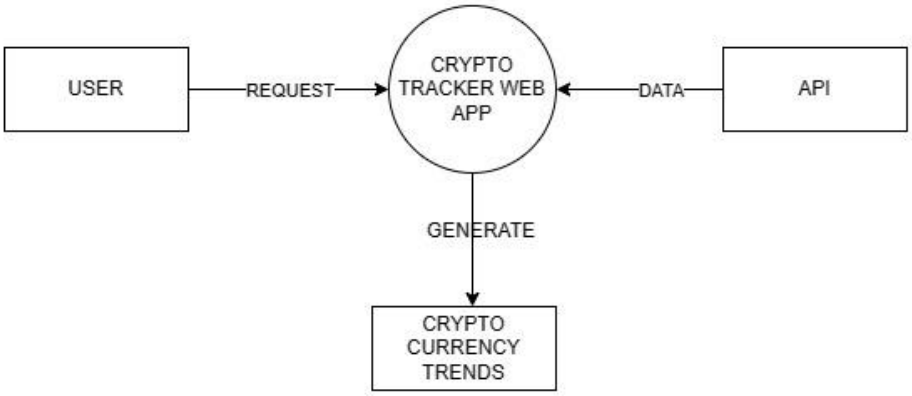
Appendix B: Analysis Models

ER Diagram

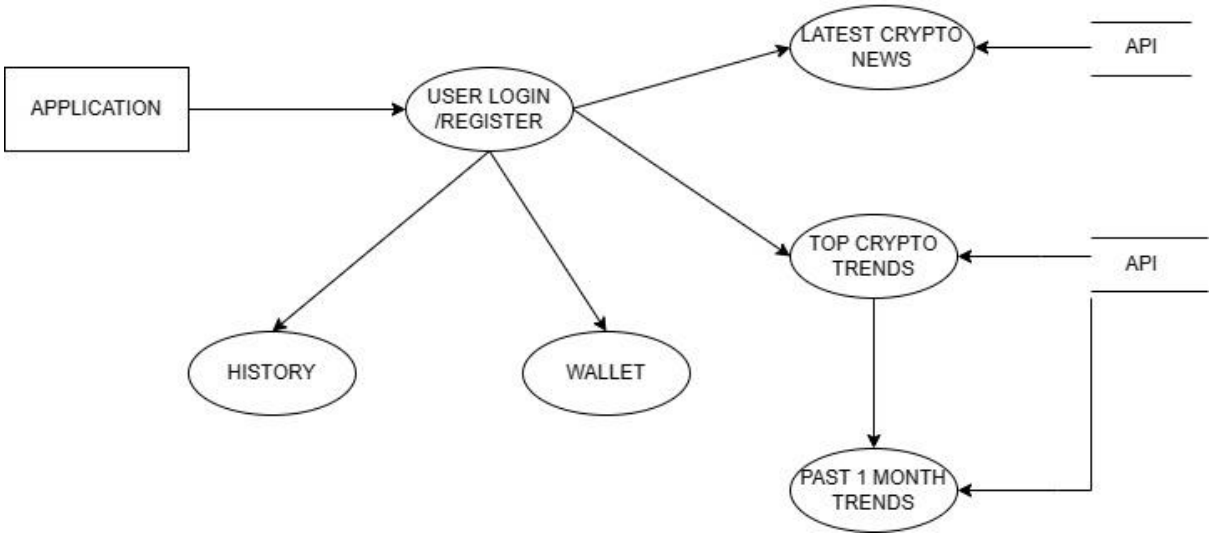


DATA FLOW DIAGRAMS (DFD)

DFD LEVEL-0

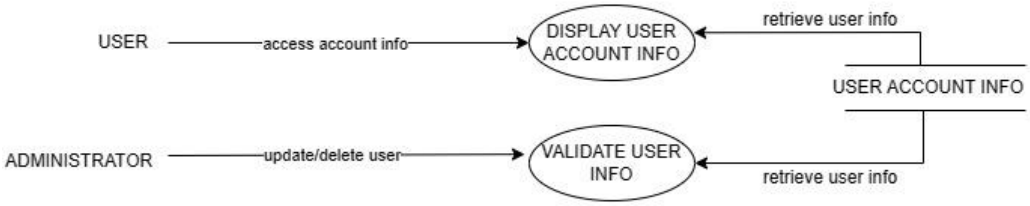


DFD LEVEL-1

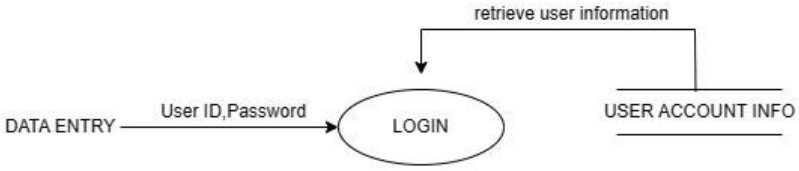


DFD LEVEL-2

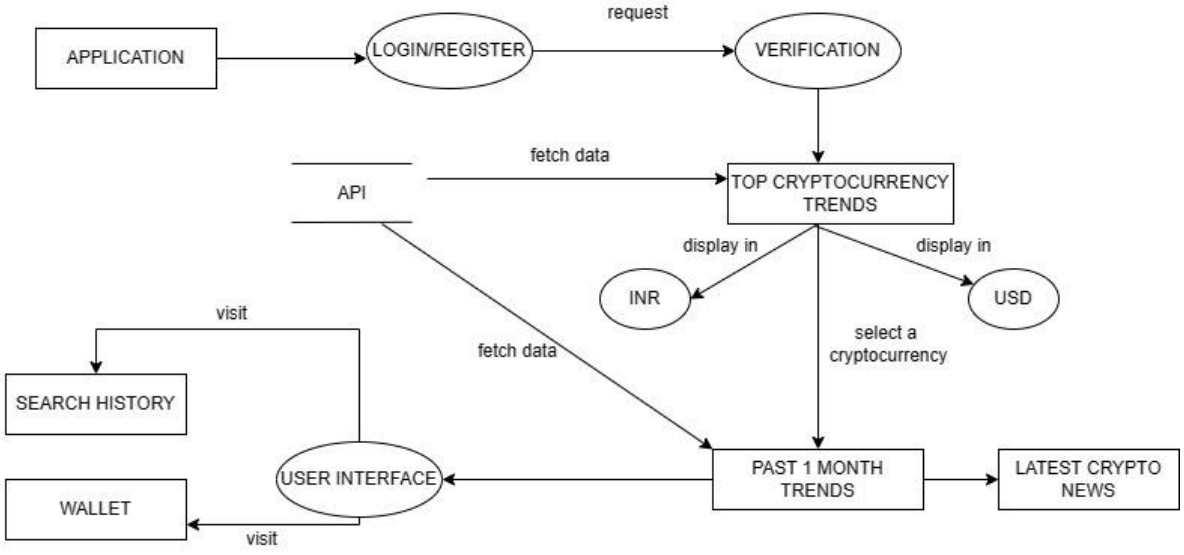
USER ACCOUNT



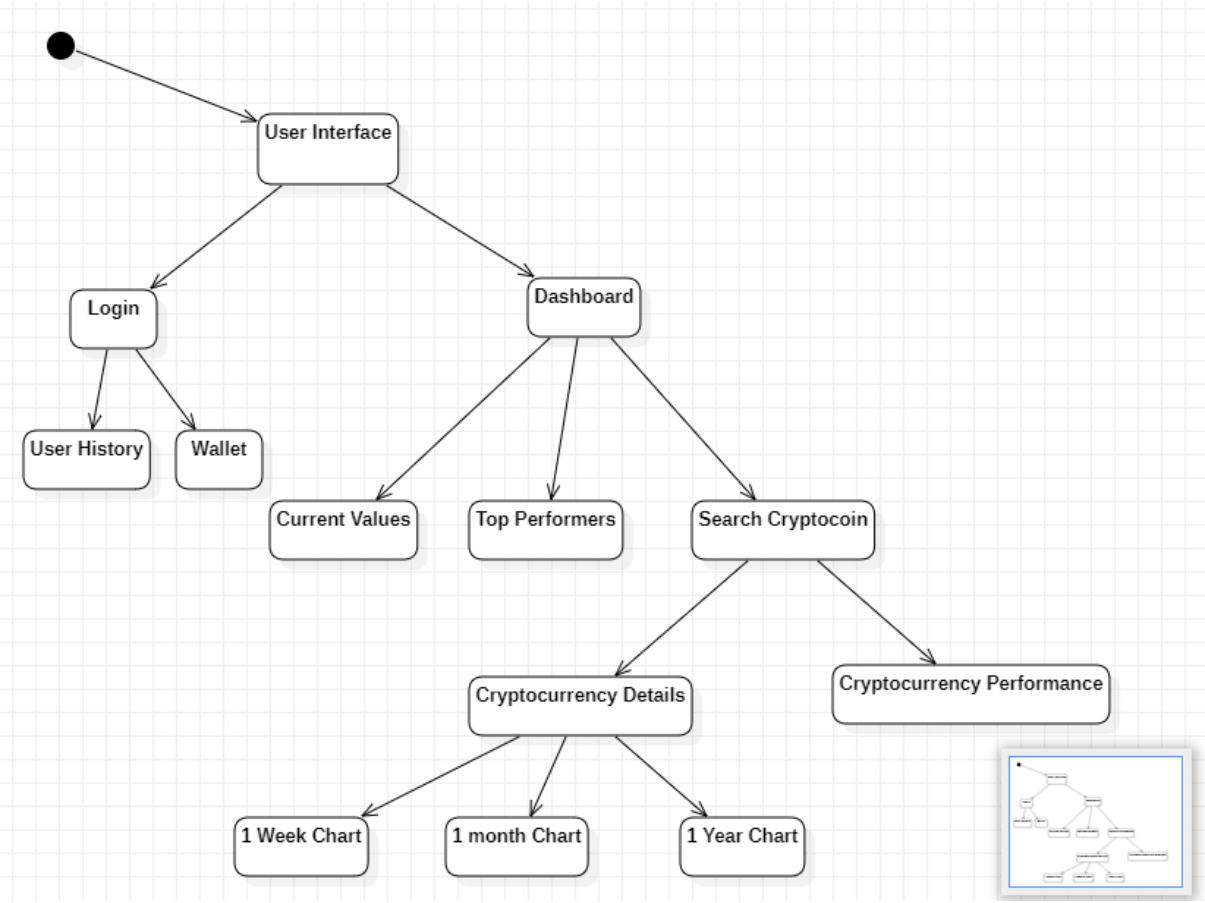
LOGIN



CRYPTOCURRENCY TRACKING



ACTIVITY DIAGRAM



Appendix C: To Be Determined List

User interface design: The design of the user interface is a crucial aspect of any web app. The app should be designed in such a way that it is easy to navigate, intuitive and visually appealing.

Supported cryptocurrencies: The webapp should support a wide range of cryptocurrencies to give users access to the most popular cryptocurrencies on the market.

News and analysis: The webapp should provide users with the latest news and analysis on the cryptocurrency market to help them make informed decisions.

Customization options: Users should be able to customize the webapp to suit their preferences, such as choosing a theme, setting up widgets, and selecting the cryptocurrencies they want to track.