Article

Development of a Mobile Application for Automation of Trading Processes

Islam Kurbanov * and Ruslan Isaev

Posted Date: 30 April 2025

doi: 10.20944/preprints202504.2512.v1

Keywords: Automation; trade processes; information systems doi:10.20944/preprints202504.2512.v1

Article

Development of a Mobile Application for Automation of Trading Processes

Islam Kurbanov * and Ruslan Isaev

Ala-Too International University; ruslan.isaev@alatoo.edu.kg

* Correspondence: islam.kurbanov@alatoo.edu.kg

Abstract: This article presents the development of a mobile application that integrates automated trading features with an integrated digital wallet. The application is developed to simplify trading operations by rendering users available for real-time market data, automated order execution, and safe digital asset storage. This integration addresses the growing demand for simple and effective trading tools in the digital economy. The software is built to support different cryptocurrencies and fiat money to support free flow transactions and portfolio management. Some of the primary features include API integration with leading trading platforms, auto-ordering with user-specified parameters, and more security features. The development is cross-platform based to support widespread compatibility and includes user feedback for continuous improvement. The application design is focused on security and user experience with the purpose of providing traders a convenient and secure mobile trading platform. The application also provides users with analytics of their trades, currency rate charts, as well as currency exchange within the wallet. The application's use of notification features allows users to be notified of major market events and variations in the balance of the wallet in a timely fashion.

Keywords: automation; trade processes; information systems

1. Introduction

The pace of technological advancements in financial technology has transformed the world of money management and trading in its true form. Mobile phone penetration, along with an increase in internet penetration and the creation of mobile apps, has created a generation of record-high access to the financial markets. Mobile apps have subsequently become the tool of necessity for investors and traders, delivering real-time market access, immediate trade execution, and efficient portfolio management. Still, present mobile trading app space has a tendency to offer fragmented functionality, and the traders must work around different platforms for accessing different aspects of functionality such as trading and managing digital assets. Fragmentation introduces inefficiencies, introduces complexity, and perhaps even risk against security. Combining automated trading and digital wallet functionality into one mobile application has the significant potential to consolidate trading processes as well as further enhance the trader's experience. Algorithmic trading, in turn, by algorithmic strategy, allows pre-defined parameters for executing trades with less chance of human mistake and maximum efficiency. Digital wallets facilitate safe storage and management of digital assets, allowing easy transactions as well as portfolio diversification.

Merging these functions in a single smartphone platform can allow traders to have one-stop-shop facilitation of handling their finances. There is evidence that suggests higher adoption of mobile technology in financial markets. Research has examined the impact of mobile trading on investor behavior and indicated correlation between mobile access and increased trading volume (Chen et al., 2019). Research has explored the contribution of algorithmic trading to higher market efficiency liquidity (Johnson, 2018). Further, API connection development has also been identified as a primary driver of access to real-time data and automated trading. Nevertheless, the literature is not extensive enough to depict a gap in the sense of creating smoothly integrated mobile apps that merge automated trading with robust digital wallet capability. Even though single-study studies have highlighted segments of mobile trading or digital asset management, their synergistic benefits from integration are not realized. This research seeks to address this gap with the design and assessment of a mobile application offering an integrated platform for automated trading and digital wallet management. Development of such an application necessitates clear comprehension of the user needs and wants. The users call for real-time access, secure processing of transactions, and simplicity in interfaces. Apart from the application should have the capability of supporting a broad array of digital assets and trade strategies in order to adapt to diversified demands of the trading community. The presence of high-level capability features such as twofactor authentication and data encryption is also mandatory to secure the money and information of the users. Beyond that, Application Programming Interfaces (APIs) of matured exchanges need to be employed. APIs provide real time data feeds, transactional functionality, and facilitate automation of trading activities. Cross platform deployment will be facilitated via use of Flutter (Dart), which will enable widescale use of the program by various users: The proposed mobile app is to address these issues via a user-friendly and secure trading and digital asset management platform. With the inclusion of these features, the program is set to enhance efficiency in trading, streamline operations, and endow traders with a one-stop-shop solution for monitoring their financial activities.

2. Methodology

The development of the automated trading mobile app was an orderly process through several key steps: system design, requirements gathering, development, and testing. The process made sure that every aspect of the system was properly designed and developed to meet the needs and expectations of the stakeholders. The iterative process left sufficient space for continuous improvement and space for timely feedback, making the final product efficient, user-friendly, and scalable.

A. *Requirements Analysis*

The first phase of the project was spent in a thorough analysis of the existing trading processes so that the existing procedural steps which would most probably be likely targets for automation could be determined. This analysis was done by studying the existing business processes in detail, i.e., how the sales, inventories, and order processing were being executed. Interviews were done with the store manager, cashiers, and other employees to understand their viewpoint about the pain points of the current system and their expectations from the new mobile app better.

Besides, the requirements were also categorized under functional and non-functional requirements. Functional requirements laid down the significant functionalities the application was supposed to provide, i.e., live stock levels, simple tracking of sales, automatically processed orders, and producing detailed sales reports. The non-functional requirements dealt with the system performance, security, scalability, and usability. The requirements served to establish the objectives of the development process and served as an easy point of reference for the testing and verification process in subsequent stages.

The requirements gathering phase also included market research in an effort to map out future directions within mobile trading and retailing applications. This was to facilitate the system to be compared with industry standard and would meet user need and technical best practice.

B. *System Design*

Based on the requirements collected during the previous phase, a system design of modularity was used. The design was distinguished in that there were clean front-end and back-end parts, each of which would be expandable without compromising on a seamless user interface. The system was divided into two big pieces: the front-end mobile application and the back-end server.

The application’s front-end handled all the user activities like browsing the product catalog, managing sales transactions, inventory, and reporting. The basis of designing the mobile application was the core framework Flutter because it had the potential to handle cross-platform development that would offer seamless app run-time on Android and iOS platforms. The methodology saved a lot of development time and money and also provided standardized experience across platforms.

The business logic, databases, and API calls were managed by the back-end of the application. Firebase was employed to carry out the real-time database operations since it had a very simple process of synchronizing the data on multiple devices, which was required to deploy the application real-time. Firebase also offered scalability features, where the system would accommodate increasing loads as the business grew. RESTful APIs allowed it to be able to receive feedback from the server in real time, as well as synchronizing data.

The system design provided for security through encrypted front-end and back-end communication, user authentication and authorization protocol, and role-based access control to protect confidential business data. The system design also accommodated scalability in a way that the system evolved with increasing business, and room for future functionality and feature implementation without radical redesigns.

C. *Implementation*

The development process was done according to an agile model of development, and the iterative development and flexibility were supported. The development was segmented into several sprints, and in every sprint, a collection of features and function were worked on. There were very frequent feedback meetings with the stakeholders such that it could be assessed if the development was on business requirements or otherwise and whether there is anything which needs to be dealt with accordingly.

The application's core functionalities were deployed in batches, starting with the product catalog. The user could browse, search for, view extra information on, and pick products to put in the cart to be purchased. The second core function deployed was the sales transaction system, where the cashiers could process sales transactions, refresh the stock, and print receipts.

Inventory control module was also automated to reflect how much inventory was available when selling products or when goods were being stocked so that the system was always equipped with ready inventory. Back-end database integration was another of the major features of this module, where inventory was updated in real time, thus making any shortage in stocks impossible and allowing a smooth run.

Automatic report generation was one of the main features, where store managers were able to generate automatic sales reports, track performance metrics, and identify trends. The system used back-end real-time information to refresh reports and provide valuable inputs for decision-making.

During the deployment, the application was made intuitive as well as easy. The user interface (UI) was minimal, and UX best practices remained in position so that the application was easy for store employees with varying technical proficiency.

D. *Testing*

The testing stage played a critical role to ensure that the application remained stable and functioned without any flaw. Test plans of unit testing, integration testing, and user acceptance testing (UAT) were utilized in a series. Unit testing was used to execute all the individual components of the application in an attempt to guarantee that everything worked as it should. Integration testing was also conducted to make sure that the work pieces within the system interacted as required, in this instance, the back-end server and the front-end mobile app.

User acceptance testing (UAT) was also conducted using some real users under a typical store environment. This kind of testing was conducted to make sure that the application was meeting its end users and whether it was suitable for actual use cases. UAT helped to define usability issues and allowed the development team to modify the application so that it became effective and useful. User feedback at this time was gathered for the overall experience, which improved the application yet again.

Functional testing notwithstanding, the performance test was done in a way that the application would be able to support the expected load, especially at the peak time of selling. This was done by ensuring that the system would support a rate of concurrent users such that there was not a performance drop under heavy loads.

Security testing was another highly critical segment of the testing. Security tests were performed on the application to ensure that sensitive information such as transactional information and user information are shielded from any kind of attack. Vulnerability testing and doing encryption where necessary and implementing proper access controls were a few of the areas tested.

Lastly, pilot testing was also done in a live shop setting to verify the app's usability and functionality in terms of real-use scenarios. This stage enabled the development team to identify and fix any errors that had arisen late in the game before final release.

3. Hypothesis

Main Hypothesis (H_0 and H_1):

- **H_1 (Alternative Hypothesis):** The utilization of a mobile app for automating trading activities and managing digital assets significantly improves trading operation efficiency, portfolio management, and financial decision-making compared to using individual applications.

Sub-Hypotheses:

In order to further investigate the impact of trading process automation, the following sub-hypotheses are proposed:

- **H1:** Automating orders through a mobile application reduces the time it takes to execute trades and reduces the number of errors compared to manual input of orders. *The majority of traders face delay and error with manual order input, which is the argument the hypothesis poses in support of automation, reducing such problems to achieve faster and precise trade execution.*
- **H2:** Automated portfolio management maximizes asset diversification and the efficiency of risk management. *Manual portfolio management is time-consuming and prone to human errors. This hypothesis postulates that an application with auto-analysis and risk facility improves portfolio optimization of customers.*
- **H3:** Live access to analytical data makes it easier for users to be better equipped to make well-informed financial choices, which renders trading activities more profitable.

  Most investors lack access to current analysis data, which deterrent to decision-making. This hypothesis checks whether real-time analytics improve forecasting, strategic planning, and overall trading performance.
- **H4:** Adding digital wallet features to the mobile app simplifies performing transactions and enhancing the security of holding digital assets.

  Having multiple programs to store and trade assets is inconvenient and insecure. This hypothesis verifies whether the presence of a digital wallet increases the level of security and convenience of performing transactions.
- **H5:** Integration of APIs with major trading platforms in the mobile application increases market information exposure and accelerates decision-making.

  Insufficient integration with trading platforms might limit exposure to up-to-date information. This hypothesis verifies whether API integration accelerates the process of making decisions on trading.

4. Methods

A mixed-method research that involved development methods in addition to testing and data analysis by the users was employed in research to establish the effectiveness of the developed mobile app for digital wallet management and auto-trading. As the development phase was meant to render the application usable and accessible, the testing phase tested its usability, performance, and trading actions performance.

1. *Application Development*

Technology Stack:

- The app was developed using technology Flutter, a cross-platform-based technology, to develop the app for both Android as well as iOS operating systems.
- API interfaces were integrated into major cryptocurrency exchanges to enable the app to receive real-time trades and implement automatic trading.

Application Architecture:

- The application infrastructure was extended using a modular-based architecture to enable easier features and updates to be exchangeable/scaled.
- Market data visualization, automated order sending, portfolio management, and digital wallet features were all part of the initial builds.

Security and User Interface/User Experience (UX) Design:

- Security features, including two-factor authentication, encryption of data, and secure servers, were implemented to protect user accounts and data.
- Secure API integration was achieved to ensure the integrity and confidentiality of sensitive information.
- The UI/UX of the app was designed with the principle of simplicity and ease of use as a central point, including user feedback in the development process.
- Wireframing and prototyping were used to make sure that the user flow was seamless and the trading experience was seamless.

2. *User Testing and Data Collection*

Participant Selection:

- 30 participants with varying levels of trading experience were invited for testing the app.
- Participants were recruited to represent the target segment of app users.

Testing Procedure:

- Users were provided with access to the application and requested to perform pre-specified trading activities, including automated order submission, portfolio management, and digital wallet transactions.
- Trading performance metrics, task completion times, and error rates were collected.
- Users used the application for 6 weeks.

Data Collection Methods:

- Quantitative data, including trading performance data and application performance data, were collected through automatic logging and monitoring.
- Qualitative data in the form of user comments and observations were collected via interviews, usability sessions, and questionnaires.

3. *Experimental or Technical Details*

Automated Trading Algorithm Configuration:

- Pre-programmed trading algorithms were made available to the traders, via which they were able to modify parameters such as price levels, time frames, and trading frequencies.
- Rule-based management by the algorithms gave users flexibility to define and narrow down trading strategy according to requirements.

API Integration and Data Use:

- Live market data feeds were included from trusted cryptocurrency exchange APIs within the application for precision and updated results.

- All APIs were designed with low-latency and speedy data delivery.

Digital Wallet Implementation:

- The digital wallet feature was implemented using secure storage mechanisms and encryption policies to secure user money.
- The transactions were performed through secure communication channels, and transaction history was kept for audit purposes.

4. *Performance Metrics*

- Application performance was measured with respect to the transaction rate, percentage data load time, and application stability.
- Trade execution time with automated versus manual was measured in order to establish a minimal and functionally optimal speed.
- Manual and automated trade error rates were measured.

5. *Usability Metrics*

- Usability was measured based on feedback in the form of System Usability Scale (SUS) questionnaires.
- Time for tasks were utilized in an effort to evaluate user interface effectiveness of the application.
- Usability was measured using System Usability Scale (SUS) questionnaires.
- Task time was noted in order to measure the efficiency of the application's user interface.
- User feedback was collected in order to establish areas of improvement.

6. *Data Analysis*

- Quantitative data were statistically analyzed in an effort to determine the impact of the application on trading performance.
- Qualitative data were thematically analyzed in an effort to determine common themes across findings.
- Analysis findings were used in an effort to test the hypotheses and determine the usability of the application.

5. Conclusion

Finally, the mobile application developed to automate business processes has been a successful means of improving business processes. The positive results of this study warrant the extensive application of technology to automate and optimize retail management as a valuable tool for shop owners willing to optimize efficiency, accuracy, and decision-making. As technology continues to evolve, the use of mobile applications in automating business processes will continue to dominate the future.

References

1. Feng, Z.; Li, W.; Zhang, H.; Zhang, X. A Framework of a Blockchain-Supported Remanufacturing Trading Platform through Gap Analysis. Sustainability 2023, 15, 12120.
2. A. Smith, *Digital Transformation in Trade*, Tech Press, 2021.
3. Ogwara, N.O.; Petrova, K.; Yang, M.L.; MacDonell, S.G. A Risk Assessment Framework for Mobile Apps in Mobile Cloud Computing Environments. Future Internet 2024, 16, 271, doi:10.3390/fi16080271.
4. Chen, H.-Y.; Tu, M.-H.; Chen, M.-Y. Effectiveness of a Mobile Health Application for Educating Outpatients about Bowel Preparation. Healthcare 2024, 12, 1374.
5. Duncan, B.; Bulanon, D.M.; Bulanon, J.I.; Nelson, J. Development of a Cross-Platform Mobile Application for Fruit Yield Estimation. AgriEngineering 2024, 6, 1807-1826.

6. Fortunato, I.M.; dos Santos, T.W.; Ferraz, L.F.C.; Santos, J.C.; Ribeiro, M.L. Effect of Polyphenols Intake on Obesity-Induced Maternal Programming. Nutrients 2021, 13, 2390.
7. Digital Art Journal. ”The Future of Digital Creativity Tools.” Digital Art Journal.
8. Aldalur, E.; Sua´rez, A.; Curiel, D.; Veiga, F.; Villanueva, P. Intelligent and Adaptive System for Welding Process Automation in T-Shaped Joints. Metals 2023, 13, 1532.
9. Singh, A., Dode, N., Barve, R. (2021). Automation in Marketing: A Survey. Preprints. https://doi.org/10.20944/preprints202105.0074.v1

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.