# Тест-план проекта SI-Wallet

1. **Test Item**

   SI-Wallet – TypeScript-powered online project with crossplatform mobile app and server side, based on React Native and NestJS to store all your coins and tokens in a single, secure mobile wallet.

   The mobile application is implemented using the React Native framework, which allows you to write cross-platform applications on different operating systems with the ability to inject the native code of each platform.

   Server implemented by using NEST JS framework with help of Sequelize ORM technology. Server based on REST API and JWT-strategy authentication.

   The system should be fast and flexible for instant deployment to any device and the ability to quickly change existing implementations during further development.

   Since the main logic for working with products is on the server, clients a will be used as server testing, which can be used after the server is started. Client testing will be done manually.

   The functional requirements described in the SRS were not fully implemented because of limitations in time and limited functionality of Binance API, but the main functionality of the SI-Wallet as a crypto wallet was fulfilled.

   Non-functional requirements are almost completely implemented, everything is implemented, except for the coverage of the unit testing code on front-end.

2. **Risk Issues**

   Despite the fact that our breath-taking wallet has no vulnerabilities - deposit money for your own risk 'cause of usage of external blockchain API block.io.

3. **Features to be tested**

   A series of tests will be carried out for this system, which will include testing the following parts of the application: authorization, work with news, cryptocurrencies and transactions.

4. **Test Approach**

   Так как основная логика работы с товарами находится на сервере, то в качестве тестирования сервера будут использоваться клиенты и среда тестирования GraphQL, которую можно использовать после запуска сервера. Тестирование клиентов будет проходить в ручном режиме.

5. **Pass/ Fail Criteria**

   The results will be presented in the form of a table containing a description of the test and its result. The test can be considered passed only if the actual result completely coincides with the expected one, otherwise the test is considered a failed one. Login, registration, display of news and cryptocurrencies, and execution of transactions will be tested.

6. **Conslusion**

The above requirements allow you to test the operation of the system and make sure it is stable, which will help to develop further plans for the development of the system and its introduction to the market.