

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені Тараса Шевченка
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра програмних систем і технологій

Дисципліна
«АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО
ЗАБЕЗПЕЧЕННЯ»

Лабораторна робота № 5
«ФОРМУВАННЯ БАЧЕННЯ ПРОГРАМНОГО ПРОДУКТУ ТА
СТВОРЕННЯ СПЕЦИФІКАЦІЯ ВИМОГ»
на тему:
«Автоматизована система безготівкових електронних платежів»

Виконав:	Гоша Д.О	Перевірів:	Юрчук І. А.
Група	ІПЗ-23	Дата перевірки	
Форма навчання	Денна	Бали	
Спеціальність	121		
2021			

Мета: Дослідити процес створення документу опису первинних вимог до програмного забезпечення та набути практичних навичок у виділенні та документуванні вимог.

Завдання:

1. Ознайомитися з особливостями створення специфікації вимог до програмного забезпечення.
2. Дослідити предметну галузь, задану згідно з варіантом та виділити функціональні та нефункціональні вимоги до програмного забезпечення.
3. Ознайомитися із базовим змістом стандартів.
4. Створити специфікацію функціональних вимог до програмного забезпечення на основі шаблону в залежності від обраної методології розробки ПЗ та відповідного формату представлення специфікації вимог
 - a. «Технічне завдання», ГОСТ 19.XXX.
 - b. «Технічне завдання», ГОСТ 34.XXX.
 - c. Software Requirements Specification, стандарт IEEE 830.
 - d. Документ Бачення в RUP
 - e. Документ Бачення в MSF. Ознайомитися з особливостями створення специфікації вимог до програмного забезпечення.

Виконання завдання:

1. Призначення

Ринок електронних гаманців рухається шляхом створення універсального засобу платежу. Провайдери вбудовують платіжні платформи в різні послуги та програми - соціальні мережі, послуги спільних поїздок, оплати рахунків, взаєморозрахунків між фізособами, ігри і т.д. Масштабне використання у різний контекст робить здійснення платежів звичайним і майже непомітним клієнтам процесом. Перевагами від використання цифрових гаманців користуються не лише фізособи, а й бізнес. Для рітейлу вони стають аналогом інтегрованих платежів та ключовим елементом бізнес-процесів у роздрібних продажах. Аналітичні алгоритми дозволяють виявляти нові тенденції у споживчому попиті та створювати більш персоналізовані продукти та послуги. «Електронні гаманці як в Україні, так і в інших країнах є одним з найдоступніших та найпопулярніших способів оплати. Число онлайн-користувачів та їх активність у цифровому середовищі продовжує зростати. Особливо це виявилось у «карантинні» місяці. По всьому світу активно зростають небанківські платіжні послуги, задаючи довгостроковий тренд. «Електронні гаманці здобули величезну популярність завдяки своїй простоті та доступності. Вони підвищують проникнення фінансових послуг навіть у віддалених регіонах і є драйвером розвитку певних ринків – зокрема, електронної комерції, завдяки своїй зручності та безпеці. У перспективі в електронних гаманцях все більше застосовуватимуться технології штучного інтелекту та криптографії, що дозволить споживачам краще керувати витратами та захищати свої фінанси», – сказав Мартін Стіллер, аналітик фінансового сектору, IDC Financial Insights.

1.1. Система електронних платежів Національного банку України (шифр СЕП-4) Загальні правила реалізації стандарту ISO 20022, спільні для всіх інструментів.

1.2. Міждержавні й національні стандарти типографії:

1.2.1. Знаки для розмітки оригіналів і виправлення коректурних і пробних відбитків. Загальні вимоги (відповідно до ГОСТ 7.62-90)

1.2.2. 2. Видання. Основні види. Терміни та визначення (відповідно до ДСТУ 3017-95)

1.2.3. . Видання. Поліграфічне виконання. Терміни та визначення (відповідно до ДСТУ 3018-95)

1.2.4. Технологія поліграфічних процесів. Терміни та визначення понять (відповідно до ДСТУ 3003:2006)

1.2.5. Скорочення слів в українській мові у бібліографічному описі.

1.3. Зміст (скорочено)

1. Призначення	3
2. Загальний опис.....	5
3. Функціональність системи.....	7
4. Вимоги до зовнішніх інтерфейсів	7
5. Інші нефункціональні вимоги Стандартизація	8
6. Висновки.....	9

- 1.4. Відповідно до ГОСТ 34.601-90 "Автоматизовані системи. Стадії створення" , після етапу формування (виявлення) вимог до системи виконується етап розробки концепції системи. Тому я бачу такі цілі:
- 1.4.1. Довгострокова підтримка продукту, з платними оновленнями, якщо це стосується нового функціоналу.
 - 1.4.2. Розробка продукту кінчається після етапу (Transition) , який включає в себе фінальну версію продукту та передається від розробника до замовника. Це включає програму бета-тестування, навчання користувачів, а також визначення якості продукту. Якщо якість не відповідає очікуванням користувачів або критеріям, встановленим у фазі Початок, фаза впровадження повторюється знову. Виконання всіх цілей означає досягнення віхи готового продукту (Product Release) та завершення повного циклу розробки.
- 1.5. Корисні посилання
- 1.5.1. <https://tradersunion.com/ru/interesting-articles/exodus-koshelek/>
 - 1.5.2. https://bank.gov.ua/files/ISO20022/Pravula.identifikazii.NBU.ver.2.2_28.09.21.pdf
 - 1.5.3. https://openarchive.nure.ua/bitstream/document/12371/1/Dovidnik_DS_TU_VPS.pdf
 - 1.5.4. <https://profinvestment.com/exodus/>
 - 1.5.5. https://redeemcdn.sfo2.digitaloceanspaces.com/public/2020_chainalysis_is_geography_of_crypto_report.pdf
 - 1.5.6. https://www.cbr.ru/Content/Document/File/132241/Consultation_Paper_20012022.pdf
 - 1.5.7. <https://pandia.ru/text/80/443/24789.php>
 - 1.5.8. https://ru.wikipedia.org/wiki/Rational_Unified_Process
 - 1.5.9. <https://intellect.icu/rational-unified-process-rup-kak-metodologiya-razrabotki-programmnogo-obespecheniya-6876>
 - 1.5.10. <https://github.com/Exodus-Privacy/exodus/blob/v1/doc/install.md#installing-your-development-environment>
 - 1.5.11. https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%B0_%D0%B4%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D1%96%D1%8F
 - 1.5.12. <https://habr.com/ru/post/231961/>

2. Загальний опис

- 2.1. Це анонімний мультикриптовалютний HD-гаманець для зберігання та виконання операцій із віртуальними монетами. Десктопна програма призначена для встановлення на пристрої під керуванням операційних систем: Windows, MacOS, Linux. Мобільний додаток (Android, iOS). Криптогаманець підтримує зберігання найбільш потрібованих та ліквідних

монет. Він повинен підтримувати 100+ криптовалют та токенів: Bitcoin, Ethereum, Bitcoin Cash, Litecoin, TRON, Dash, Binance Coin, Ethereum Classic, EOS, Zcash та інших.

2.2. Базовий функціонал (не включаючи інтуїтивно зрозумілі речі типу переказу грошей або прийому)

2.2.1. Кількість підтримуваних криптовалют понад 150.

2.2.2. Мобільний додаток.

2.2.3. Десктопна програма.

2.2.4. Комісії за зберігання.

2.2.5. Верифікація.

2.2.6. Обмін криптовалют між собою.

2.3. За статистикою Chainalysis Україна займає перше місце за показником

Country	Score	Rank	Rank of individual weighted metrics feeding into index			
			On-chain value received	On-chain retail value received	Number of on-chain deposits	P2P exchange trade volume
Ukraine	1	1	4	4	7	11
Russia	0.931	2	7	8	5	9
Venezuela	0.799	3	19	14	15	2
China	0.672	4	1	1	95	53
Kenya	0.645	5	37	11	57	1
United States of America	0.627	6	5	6	39	16
South Africa	0.526	7	12	9	41	10
Nigeria	0.459	8	14	7	112	3
Colombia	0.444	9	25	18	61	4
Vietnam	0.443	10	2	2	44	81

Global
Crypto
Adoption
Index

Це
свідчи
ть про
великий
рівень

інтеграції цифрових валют у країні. Україна надіслала \$8,2 млрд дол. і отримав 8,0 мільярдів доларів. Хоча ці цифри набагато нижчі, ніж суми, які ми бачимо Китай, США та інші провідні країни вказують на набагато вищий рівень прийняття якщо врахувати чисельність населення та економіки країни. Наш індекс також факторів у двох інших показниках мережі, а також обсягів однорангової торгівлі (P2P). Білорусь, наприклад, посіла третє місце в регіоні та 19 місце в цілому значною мірою завдяки своєму сильному P2P діяльності. Україна не займає перше місце за чотирма індивідуальними показниками індексу, але вони перевершують багато країн за стабільним рейтингом серед лідерів від десяти до одинадцяти країн для цих показників.

Country	Score	Rank	Rank of individual weighted metrics feeding into index			
			On-chain value received	On-chain retail value received	Number of on-chain deposits	P2P exchange trade volume
Ukraine	1	1	4	4	7	11
Russia	0.931	2	7	8	5	9
Belarus	0.241	19	42	44	25	12

Аудиторія переважно чоловіки віком від 18 до 40 років.

2.4. Середовищами функціонування продукту будуть усі відомі ОС системи (десктоп) та мобільні. Перераховуються нижче:

2.4.1. Десктоп (Windows, MacOS, Linux)

2.4.2. Мобільні (Android, iOS)

Місце знаходження користувачів різняться рівнем законодавчого

визнання електронних валют та легалізації або декриміналізації технології як такої.

Географічне знаходження серверів і компанії буде визначатися розміром оподаткування доходів і залежитиме від законодавчого рівню. Не виключно що економічно доцільно буде розмістити компанію у офшорній зоні, яка спрямована на мінімізацію податків у сфері електронних платіжних систем. Сервер слід розміщати у країнах з більш м'якою правовою системою і великою кількістю даних центів з низькооподаткованими відшкодуваннями персоналу.

2.5. Технології, що використовуватимуться під час розробки: Postgresql 13 ,

2.5.1. Rabbitmq server, Essential, Libssl, Dexdump, Libffi, python, Minio server.

2.5.2. Середовище розробки – pycharm, Docker.

2.5.3. Залежності – libxml2-dev, libxslt1-dev, libpq-dev, pipenv.

2.6. ГОСТ 19.101-77 (Експлуатаційні документи)

Містить відомості для забезпечення функціонування та експлуатації програми. Виконуються на стадії робочого проєкту. Необхідність — за погодженням для кожного документа окремо. Види експлуатаційних документів:

2.6.1. Формуляр (код виду документа — 30) — містить основні характеристики програми, комплектність і відомості про експлуатацію програми,

2.6.2. Опис застосування (код виду документа — 31) — містить відомості про призначення програми, області застосування, застосовуваних методах, класі вирішуваних задач, обмеження для застосування, мінімальної конфігурації технічних засобів,

2.6.3. Керівництво системного програміста (код виду документа — 32) — містить відомості для перевірки, забезпечення функціонування та налаштування програми на умови конкретного застосування,

2.6.4. Керівництво програміста (код виду документа — 32) — містить відомості, необхідні для експлуатації програми,

2.6.5. Керівництво оператора (код виду документа — 33) — містить відомості для забезпечення процедури спілкування оператора з обчислювальною системою в процесі виконання програми,

2.6.6. Опис мови (код виду документа — 35) — містить опис синтаксису і семантики мови програмування,

2.6.7. Керівництво з технічного обслуговування (код виду документа — 46) — містить відомості для застосування тестових і діагностичних програм при обслуговуванні технічних засобів. Склад застосовуваних видів документів визначається розробником документів залежно від стадій розробки технологічної документації та типу виробництва.

2.7. Залежностей немає.

3. Функціональність системи

3.1. Функціональні вимоги з коротким описом.

- 3.1.1. Обмін криптовалютами між собою (високий пріоритет).
- 3.1.2. Кількість підтримуваних криптовалют понад 150 (середній пріоритет).
- 3.1.3. Мобільний додаток (високий пріоритет).
- 3.1.4. Десктопна програма (високий пріоритет).
- 3.1.5. Комісії за зберігання (низький пріоритет).
- 3.1.6. Верифікація (низький пріоритет).

4. **Вимоги до зовнішніх інтерфейсів**

1061-1998 IEEE Standard for Software Quality Metrics Methodology ISO
8402:1994 Quality management and quality assurance

- 4.1. Вимоги до повторного використання реалізації або компонентів програми або системи (Reusability). Поки обмежимося лише тим, що найчастіше ці вимоги виникатимуть там, де загальні компоненти використовуються декількома модулями системи, що розробляється.
- 4.2. Вимоги до розширення (Extensibility) програми або системи у зв'язку з появою нових функціональних вимог, тісно пов'язане з таким архітектурним атрибутом якості, як переносимість коду. Як правило, на початковому етапі збору вимог можна обмежитися вказівкою тих функціональних областей, які надалі повинні задовольняти вимогу розширюваності.
- 4.3. Вимоги до міграції (Portability) програми або системи на інші платформи.
- 4.4. Вимоги до взаємодії між компонентами рішення, між зовнішніми компонентами, використання стандартних протоколів та технологій взаємодії (Interoperability). Наприклад, до таких вимог можна віднести можливість використання декількох стандартних протоколів для обміну даними між однією з підсистем системи, що розробляється, і зовнішньою системою-постачальником даних (наприкладі ArcGIS).
- 4.5. Вимоги до підтримки системи або програми (Supportability). Серед цих параметрів можуть бути названі такі як, наприклад, дешевизна та швидкість розробки, прозорість поведінки програми, простота аналізу помилок та проблем у роботі.
- 4.6. Вимоги до модульності програми або системи (Modularity). Зазвичай такі вимоги вказують, як система має бути розділена на модулі, або перераховують список обов'язкових модулів, які мають входити до складу системи.
- 4.7. Вимоги до можливості тестування (Testability) програми або системи визначають обсяг вимог до автоматичного та ручного тестування, наявність необхідного інструментарію.
- 4.8. Вимоги до можливості та простоти локалізації (Localizability) програми чи системи визначають можливості та специфічні архітектурні вимоги, що накладаються процесом локалізації. Ці вимоги містять також перелік мов, на які передбачається виконувати локалізацію програми чи системи.

5. Інші нефункціональні вимоги

Стандартизація - ISO 9126 , ГОСТ 34, Модель якості по МакКоллу (McCall's Quality Model) , Модель якості по Боєму (Boehm's Quality Model)

- 5.1.1. Доступність — це атрибут якості, який визначає час безперервної роботи програми або системи. Щоб визначити цей параметр, зазвичай вказують максимально допустимий час простою системи.
- 5.1.2. Надійність - вимога, що описує поведінку програми або системи в позаштатних ситуаціях (приклади: автоматичний перезапуск, відновлення роботи, збереження даних, дублювання важливих даних, резервування логіки) Вимоги до часу зберігання даних (наприклад, використання БД як постійне сховище даних, тривалість зберігання даних)
- 5.1.3. Масштабованість — вимоги до горизонтального та/або вертикального масштабування програми або системи. Говорячи про вертикальну масштабованість, ми визначаємо вимоги до вертикальної архітектури системи або програми. До вимог вертикальної масштабованості можуть належати, наприклад, можливість перенесення додатків більш потужні SMP-системи, підтримка великого обсягу пам'яті і файлів. Говорячи про горизонтальну масштабованість, ми визначаємо вимоги до горизонтальної архітектури системи або програми. До вимог горизонтальної масштабованості можуть належати, наприклад, можливість використання технологій кластеризації. Слід особливо помітити, що вертикальне масштабування зазвичай спрямовано підвищення продуктивності системи. Горизонтальне масштабування, крім продуктивності, дозволяє підвищити стійкість до відмови системи.
- 5.1.4. Вимоги до зручності використання системи/додатку (з погляду користувача) та вимоги до зручності та простоти підтримки (Usability)
- 5.1.5. Вимоги до безпеки, як правило, включають три великі категорії: вимоги, пов'язані з розмежуванням доступу, вимоги, пов'язані з роботою з приватними даними, та вимоги, спрямовані на зниження ризиків від зовнішніх атак.
- 5.1.6. Вимоги до конфігурованості програми, взаємодії та розташування компонентів можна умовно поділити на чотири рівні:
 - 5.1.6.1. Конфігурованість на основі наперед визначеного набору параметрів (predefined configurability), коли необхідний рівень модифікації досягається шляхом зміни значень параметрів з наперед визначеного набору;
 - 5.1.6.2. Конфігурованість на основі зумовленого набору базових об'єктів (framework constrained configurability), коли необхідний рівень модифікації досягається шляхом перекомпонування зумовленого набору процесів, сутностей та службових процедур;
 - 5.1.6.3. Конфігурованість шляхом реалізації нових базових об'єктів (basis reimplementation), коли забезпечується розширення набору процесів та сутностей;

- 5.1.6.4. Конфігурованість шляхом нової реалізації системи (system reimplementation), коли система повинна встановлюватися та налаштовуватися з нуля.
- 5.1.7. Вимоги до продуктивності рішення, що визначаються в термінах кількості одночасно працюючих користувачів, транзакцій, часу реакції, тривалості обчислень, а також швидкості та пропускну здатності каналів зв'язку
- 5.1.8. Обмеження, що накладаються на об'єм доступної пам'яті, процесорного часу, дискового простору, пропускну здатність мережі, при яких програма повинна ефективно виконувати покладені на неї завдання

6. Висновки

У цій лабораторній роботі я здобув навички у створенні специфікації функціональних вимог до програмного забезпечення на основі шаблону Software Requirements Specification, стандарт IEEE 830 методології розробки (RUP) та відповідного формату представлення специфікації вимог. Досліджено процес створення документу опису первинних вимог до програмного забезпечення та набуто практичні навички у виділенні та документуванні вимог. Виділено функціональні та нефункціональні вимоги до програмного забезпечення.