

УДК 004.9

DOI <https://doi.org/10.32689/maup.it.2021.1.5>

**Микола РУДНІЧЕНКО**

кандидат технічних наук, доцент кафедри інформаційних технологій, Державний університет «Одеська Політехніка», просп. Шевченко, 1, м. Одеса, Україна, індекс 65001 ([nickolay.rud@gmail.com](mailto:nickolay.rud@gmail.com))

ORCID: <https://orcid.org/0000-0002-7343-8076>

**Володимир ВИЧУЖАНІН**

доктор технічних наук, професор, завідувач кафедри інформаційних технологій, Державний університет «Одеська Політехніка», просп. Шевченко, 1, м. Одеса, Україна, індекс 65001 ([vint532@yandex.ua](mailto:vint532@yandex.ua))

ORCID: <https://orcid.org/0000-0002-5244-5808>

**Наталія ШИБАЄВА**

кандидат технічних наук, доцент кафедри інформаційних технологій, Державний університет «Одеська Політехніка», просп. Шевченко, 1, м. Одеса, Україна, індекс 65001 ([nati.shibaeva@gmail.com](mailto:nati.shibaeva@gmail.com))

ORCID: <https://orcid.org/0000-0002-7869-9953>

**Ігор ПЕТРОВ**

доктор технічних наук, професор кафедри морських перевезень національного університету «Одеська морська академія», вул. Дідріхсона, 8, м. Одеса, Україна, індекс 65029 ([firmness@list.ru](mailto:firmness@list.ru))

ORCID: <https://orcid.org/0000-0002-8740-6198>

**Олександр МАЗУРЕНКО**

Студент Міжрегіональної Академії управління персоналом, вул. Фрометівська, 2, м. Київ, Україна, індекс 03039 ([alexmazur@gmail.com](mailto:alexmazur@gmail.com))

ORCID: <https://orcid.org/0000-0002-3320-1348>

**Mykola RUDNICHENKO**

PhD in Technical Sciences, Associate Professor of Information Technology, Odessa Polytechnic State University, ave. Shevchenko, 1, Odessa, Ukraine, postal code 65001 ([nickolay.rud@gmail.com](mailto:nickolay.rud@gmail.com))

**Vladimir VYCHUZHANIN**

Doctor of Technical Sciences, Professor, Head of the Department of Information Technologies, Odessa Polytechnic State University, ave. Shevchenko, 1, Odessa, Ukraine, postal code 65001 ([vint532@yandex.ua](mailto:vint532@yandex.ua))

**Natalia SHIBAYEVA**

Candidate of Technical Sciences, Associate Professor of Information Technology, Odessa Polytechnic State University, ave. Shevchenko, 1, Odessa, Ukraine, postal code 65001 ([nati.shibaeva@gmail.com](mailto:nati.shibaeva@gmail.com))

**Igor PETROV**

Doctor of Technical Sciences, Professor of the Department of Maritime Transport of the National University «Odessa Maritime Academy», str. Didrichson, 8, Odessa, Ukraine, postal code 65029 ([firmness@list.ru](mailto:firmness@list.ru))

**Alexander MAZURENKO**

Student of the Interregional Academy of Personnel Management, str. Frometivska, 2, Kyiv, Ukraine, postal code 03039 ([alexmazur@gmail.com](mailto:alexmazur@gmail.com))

**Бібліографічний опис статті:** Рудніченко М., Вичужанін В., Шибаяєва Н., Петров І., Мазуренко О. Розробка проекту кроссплатформеної розподіленої інформаційної системи прототипування зовнішнього вигляду програмних застосувань. *Інформаційні технології та суспільство*. 2021. Вип. 1. С. 43–50. DOI: <https://doi.org/10.32689/maup.it.2021.1.5>

**Bibliographic description of the article:** Rudnichenko, M., Vychuzhanin, V., Shybaieva, N., Petrov, I., Mazurenko, O. (2021). Rozrobka proektu krossplatformenoї rozpodilenoї informatsiinoї systemy prototypuvannia zovnishnoho vyhliadu proqramnykh zastosuvan [Crossplatform distributed information system of prototyping software applications interface project]. *Informatsiini tekhnologii ta suspilstvo – Information technology and society*, 1, 43–50. DOI: <https://doi.org/10.32689/maup.it.2021.1.5>

## РОЗРОБКА ПРОЕКТУ КРОСПЛАТФОРМЕНОЇ РОЗПОДІЛЕНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПРОТОТИПУВАННЯ ЗОВНІШНЬОГО ВИГЛЯДУ ПРОГРАМНИХ ЗАСТОСУВАНЬ

**Анотація.** Останні роки все більшої важливості набуває завдання превентивного узгодження та обговорення специфіки реалізації проєктів програмних застосувань та систем з замовниками, що стає можливим на базі створення структурованих та гнучких прототипів інтерфейсів користувачів. **Метою** статті є опис особливостей проєктування інформаційної системи прототипування інтерфейсу програмних застосувань. Реалізація поставленої мети передбачає вирішення низки **завдань**: 1) аналізу ключових концепцій та підходів до створення прототипів інтерфейсів програмних систем; 2) розробки схеми головного вікна інформаційної системи; 3) створенні об'єктної моделі системи на базі використання діаграм в нотації мови UML. **Наукова новизна.** У статті методика розробки прототипів інтерфейсів користувачів є розділеною на окремі архітектурні шаблони програмних застосувань, зокрема на десктопні, мобільні та веб-орієнтовані системи, що забезпечує цільовий підхід до використання компонентів та елементів інтерфейсу. Як **висновок**, у статті наголошується, що розробка прототипів інтерфейсів користувачів для програмних застосувань та систем різної направленості та прикладної сфери є актуальним та затребуваним на практиці завданням, вирішення якого дозволяє забезпечити більш чітке та послідовне розуміння загального бачення проєкту замовником та командою розробників. Створені об'єктні моделі роботи системи дозволяють виявити її архітектурні переваги та особливості, що полягають у гнучкості розподілу функціоналу та сутностей об'єктів. Наведені результати проєктування інформаційної системи є основою для її програмної імплементації засобами сучасних мов програмування високого рівня, зокрема, мовою Javascript. Подальшим шляхом розвитку системи може стати більш цільовий вектор впровадження у її архітектуру технологій розподілених систем та паралельних обчислень, що дозволить забезпечити комфортне її використання у асинхронному режимі на базі мікросервісного підходу.

**Ключові слова:** проєктування інформаційних систем, об'єктно-орієнтоване проєктування, прототипування інтерфейсу користувача

## CROSSPLATFORM DISTRIBUTED INFORMATION SYSTEM OF PROTOTYPING SOFTWARE APPLICATIONS INTERFACE PROJECT

**Abstract.** In recent years, the task of preventive coordination and discussion of the specifics of the implementation of software applications and systems projects with customers has become increasingly important, which is possible through the creation of structured and flexible prototypes of user interfaces. **The aim** The purpose of the article is to describe the design features of the information system for prototyping the interface of software applications. Realization of the set purpose provides the decision of a number of **tasks**: 1) the analysis of key concepts and approaches to creation of prototypes of interfaces of software systems; 2) development of the scheme of the main window of the information system; 3) creating an object model of the system based on the use of diagrams in UML notation. **Scientific novelty.** In the article, the method of developing prototypes of user interfaces is divided into separate architectural templates of software applications, in particular desktop, mobile and web-oriented systems, which provides a targeted approach to the use of components and elements of the interface. In **conclusion**, the article emphasizes that the development of prototypes of user interfaces for software applications and systems of various directions and applications is a relevant and popular task, the solution of which allows to provide a clearer and more consistent understanding of the overall vision of the project by the customer and development team. The created object models of the system allow to reveal its architectural advantages and features, which consist in the flexibility of the distribution of the functionality and essences of the objects. These results of information system design are the basis for its software implementation by means of modern high-level programming languages, in particular, Javascript. A further target of the system development may be a more targeted vector of introduction into its architecture of distributed systems technologies and parallel computing, which will ensure its comfortable use in asynchronous mode based on the microservice approach.

**Key words:** information systems design, object-oriented design, user interface prototyping

**Актуальність проблеми.** У розробці сучасного програмного забезпечення (ПЗ) ключовим фактором виробничого процесу є проєктування. Даний процес створення проєкту системи часто підрозділяється на 2 окремі частини: проєктування функціоналу і інтерфейсу [1].

Для проєктування функціоналу використовуються існуючі нотації UML і IDEF0, є фактично промисловими стандартами при розробці ПЗ. У проєктуванні графічного інтерфейсу користувача (GUI) в даний час відсутні усталені стандарти, проте існують окремі рекомендації, закони дизайну, прийоми, традиції, різні умови експлуатації ПЗ [2]. При цьому важливою частиною цього процесу є прототипування або макетування, тобто операція, яка полягає в створенні макета або прототипу майбутньої системи.

Макети в практиці бувають: паперовими, презентаційними, імітаційними та ін. Велика частина сучасних інтегрованих середовищ програмування (IDE) дозволяє розробляти деякі типи макетів, однак це сильно залежить від знань і навичок у використанні конкретної середовища і мови програмування. При цьому в разі створення великого проекту GUI реалізується окремим фахівцем, який не завжди повинен бути фахівцем в області розробки програмного коду. У зв'язку з цим на практиці розробнику зручно мати гнучкий інструмент для прототипування GUI, що підтримує можливості швидкого створення складних макетів, насичених елементами та об'єктами [3].

Більшість існуючих програмних продуктів, які застосовуються для вирішення завдань створення GUI, не є спеціалізованими і функціональними інструментами прототипування, однак дозволяють створювати прийнятні макети різного ступеня складності.

Останнім часом у зв'язку з підвищенням вимог до розробки інтерфейсів проявляються тенденції використання спеціалізованого ПЗ, націленого на створення прототипів GUI. Подібні системи підтримують створення прототипів для різних видів ПО: десктопних, веб і мобільних додатків [4].

Перевагами створення прототипу GUI є [5]:

- можливість побачити і протестувати інтерфейс програми, оцінивши зручність взаємодії його частин без необхідності програмування всього функціоналу, що знижує витрати по зміні GUI (чим швидше інтерфейс буде приведений до підсумкового виду, тим дешевше вийде підсумковий програмний продукт);

- наочна демонстрація можливостей замовнику;
- перевірка зручності використання GUI;
- тестування нестандартних підходів до зовнішнього вигляду інтерфейсу.

Саме прототипування GUI користувача дозволило сформуватися тенденції по відділенню розробки зовнішнього вигляду програми від безпосереднього програмування всього функціоналу. Фактично, це означає, що здійснюється не розробка дизайну інтерфейсу, на базі чого далі йде його реалізація, а паралельна розробка функціональної та графічної складової проекту [6].

Все це обумовлює актуальність проведення досліджень в даній області.

**Аналіз останніх досліджень і публікацій.** Прототип являє собою узагальнену модель, прообраз кінцевого ПО [7]. Прототипи найчастіше розрізняються за ступенем точності і близькості до підсумкового ПО, створюваному командою розробників або одним програмістом. Різні види прототипів GUI використовуються для різних цілей і призначені для вирішення різних завдань [8]. За стадії готовності прототипи поділяються на 3 етапи: концептуальні, інтерактивні і анімовані.

Прототипи надають зручну можливість для розробників не тільки для залучення в процес дизайну потенційних користувачів, а також і для оперативного створення потрібного ПО, відповідного очікуванням клієнта [9].

Прототипи ПО дозволяють спростити процес обговорення з замовником деталей реалізації, є наочною картиною для програмістів і здатні уявити ідею компанії [10].

Концептуальний прототип (КП) є структурованим схематичне відображення майбутніх екранів (форм) і розробляється на ранніх етапах проектування ПО [11].

КП слід робити завжди в разі розробки GUI нової програми. Це обумовлено тим, що подібний спосіб дозволить на ранніх стадіях вирішити більшу частину проблем з використанням функціоналу програм.

КП підходить для оперативного тестування ідей в рамках команди, що пов'язано з підтримкою швидкого розміщення основних елементів форм екранів. При цьому для розробки подібного прототипу не потрібно володіти навичками роботи зі спеціалізованими інструментами, досить скористатися підручними засобами. КП є незамінним в тих випадках, коли потрібно перенести призначені для користувача сценарії на екрани програми [12].

Перевагою КП є можливість підтримки режиму командної роботи. На практиці часто трапляється, що в разі візуалізації функціоналу потрібна підтримка окремих фахівців у вузьких областях. КП може виглядати як результат в області мислення групи людей і є досить ефективним методом вирішення проблем щодо використання проекту та пошуку рівноваги між цілями користувачів і цілями бізнесу [13].

Інтерактивний прототип (ІП) найчастіше збирається з екранів, які пройшли стадію КП. ІП стає наочним і реалістичним для подальшого його тестування на кінцевих користувачів [14].

Анімований прототип (АП) є самим високорівневим прототипом з усіх існуючих. Деякі з АП здатні досить точно моделювати роботу ПО. Ключовою перевагою розробки анімованого прототипу відрізняється рівнем інтерактивності. На даному етапі дизайнер активно працює, розробляючи аспект UX – взаємодія програми з користувачем, специфіка візуалізація [15].

Анімація є методом комунікації ПО з користувачем. Вона дозволяє користувачеві активно слідувати за відображенням подій, що в разі збільшує юзабіліті інтерфейсу. Коли руху об'єктів в програмі

моделюють природні фізичні процеси, вони зчитуються мозком на початковому (підсвідомому) рівні, а користувач не замислюючись розуміє, що відбувається. Таким чином, рух процесу є орієнтованим на користувача [16].

**Метою статті** є розробка проекту кроссплатформеної розподіленої інформаційної системи прототипування зовнішнього вигляду програмних застосувань.

**Виклад основного матеріалу.** Розробку системи доцільно почати з макетування інтерфейсу, який є основним орієнтиром при побудові функціонального складу ПЗ [14–16].

Концептуальний прототип інтерфейсу системи, що розробляється в рамках даної статті наведено на рис. 1.

У лівому верхньому куті розташований компонент головного меню системи, праворуч від нього знаходиться палітра (панель) робочих компонентів, за допомогою яких буде здійснюватися обробка і зміна властивостей окремих елементів в робочому просторі.

Зліва розташовані 3 окремі вкладки, які можуть бути розгорнуті в панельному режимі і будуть необхідні для вибору з бібліотеки і використання відповідних елементів інтерфейсу на робочому полотні (розташоване посередині вікна макета).

У правій частині розташована вкладка властивостей обраного елемента, за допомогою якої користувачеві буде надана можливість змінювати і наносити окремі параметри текстових полів, міток і інших елементів інтерфейсу.

У нижній частині розташована панель управління і подій, на якій можна буде виконувати дії зі створення нових робочих полотен і їх конфігурації.

З метою формування більш чіткого уявлення про порядок використання системи розроблений загальний алгоритм взаємодії користувача з системою.

Спочатку виконується розгортання системи і установка всіх необхідних залежностей проекту.

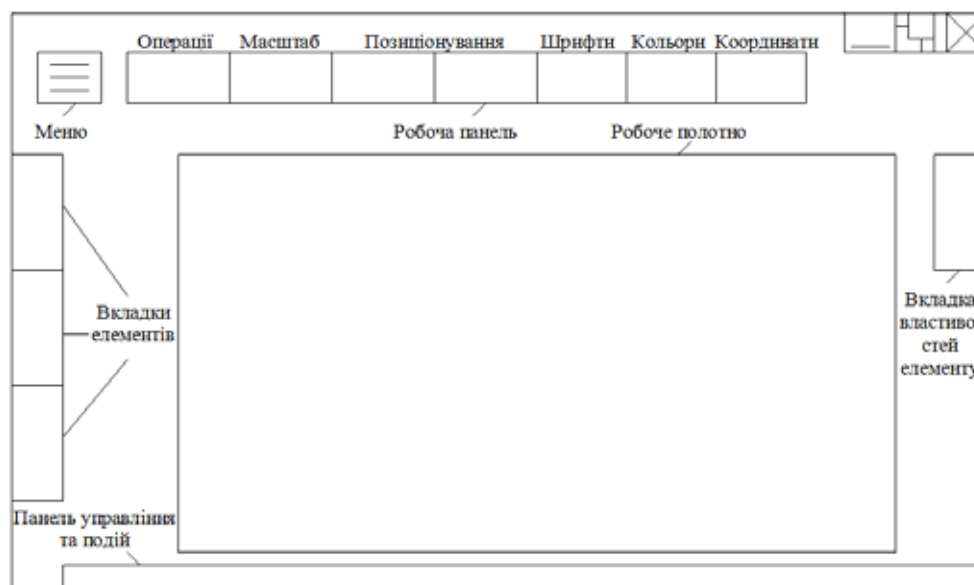


Рис. 1. Схема інтерфейсу системи

Після цього, коли система запущена на локальній станції і завантажений її інтерфейс, відбувається створення нового проекту прототипу інтерфейсу за вибором користувача.

Після цього створюється нове робоче полотно і за допомогою передбачених вище вкладок користувачем системи здійснюється пошук, вибір і компоновання елементів інтерфейсу прототипу відповідно до його завданнями в робочому просторі.

В результаті, коли всі компоненти вибрані і додані до робочого полотна здійснюється настройка їх позиціонування і вибір параметрів відображення в робочому просторі.

Якщо всі елементи відображені належним чином і якщо користувач не має більше намірів по створенню нових прототипів в проекті, то система дозволяє здійснити операції по збереженню або експорту отриманих результатів в файли відповідних розширень і закінчити роботу з програмою (закрити проект). У разі, якщо елементи відображені некоректно, користувачем здійснюється попередня коригувальна операція. У разі, якщо користувач, наприклад, проектує прототипи інтерфейсу кількох веб-сто-

рінок, він може створити відповідну кількість робочих полотен в системі і редагувати їх окремим чином. З метою детального опису функціональних можливостей створюваної системи використовуємо нотацію Use case діаграм, що дозволить розділити специфікації для розробки системи і представити їх в ієрархічному вигляді. Розроблена діаграма варіантів використання системи приведена на рис. 2.

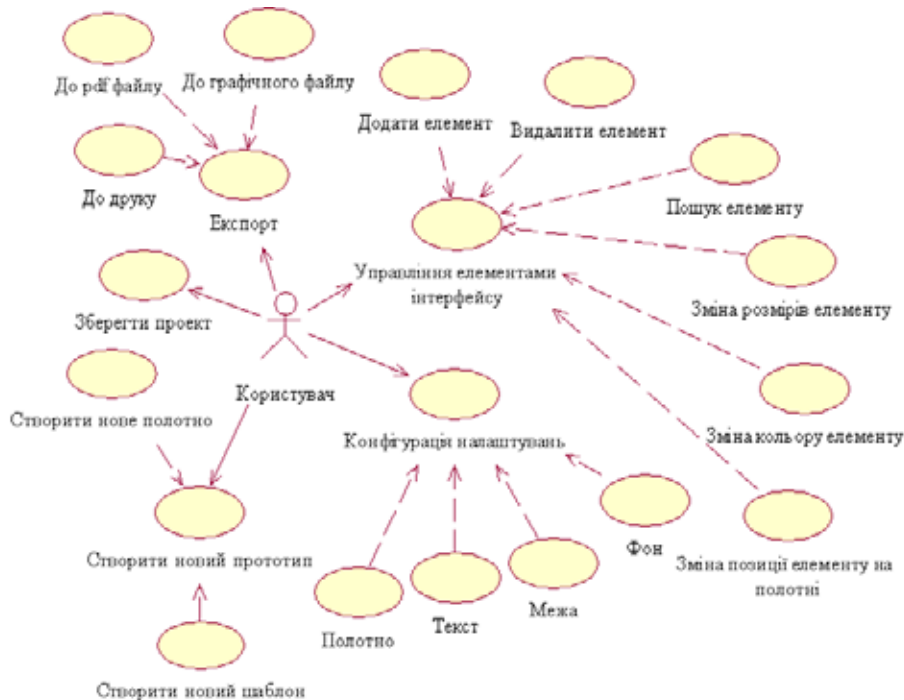


Рис. 2. Діаграма варіантів використання системи

Користувач має наступні основні можливості:

- створювати новий прототип інтерфейсу (в рамках нього він може створити нове полотно і новий шаблон);
- зберегти проект в файли відповідного типу (сериалізація даних);
- завантажити проект з файлу;
- експортувати результати на активному робочому полотні в друк (на принтер, в графічний файл \*.png і в \*.pdf формат);
- виконувати управління елементами інтерфейсу створюваного прототипу (додавати, видаляти, змінювати розміри, кольори, позиції елемента на полотні, здійснювати пошук елемента в бібліотеці);
- конфігурувати налаштування системи по окремих елементах (робочого полотна, тексту або кордону).

Фрагмент діаграми класів наведено на рис. 3. Клас Canvas відповідає за відображення елементів інтерфейсу системи, головними його нащадками є класи Exporter, Browser та Configuration, що потрібні для реалізації функціоналу з створення окремих моделей прототипів згідно до дій користувача на головній формі системи.

Оскільки склад системи планується у вигляді великої кількості компонентів (модулів Node.js, програмних файлів і бібліотек) необхідно винести на таку діаграму тільки найбільш загальні з них. Розроблена діаграма основних компонентів системи приведена на рис. 4.

Дана діаграма складається з наступних загальних компонентів:

- MainProgram – пакет головних програмних файлів реалізації системи.
- LocalHost – локальний сервер, на якому має відбуватися розгортання проекту (базується на платформі NodeJS і Electron).
- Файли CSS і HTML – для забезпечення можливостей настройки і візуалізації інтерфейсу в веб-браузері.
- UserInterface – пакет, що містить всі view-елементи системи для їх коректної взаємодії.
- Resources – пакет, що містить графічні елементи для побудови прототипів інтерфейсу.
- Bootstrap – набір інструментів і стилів для верстки та розробки веб-додатків.

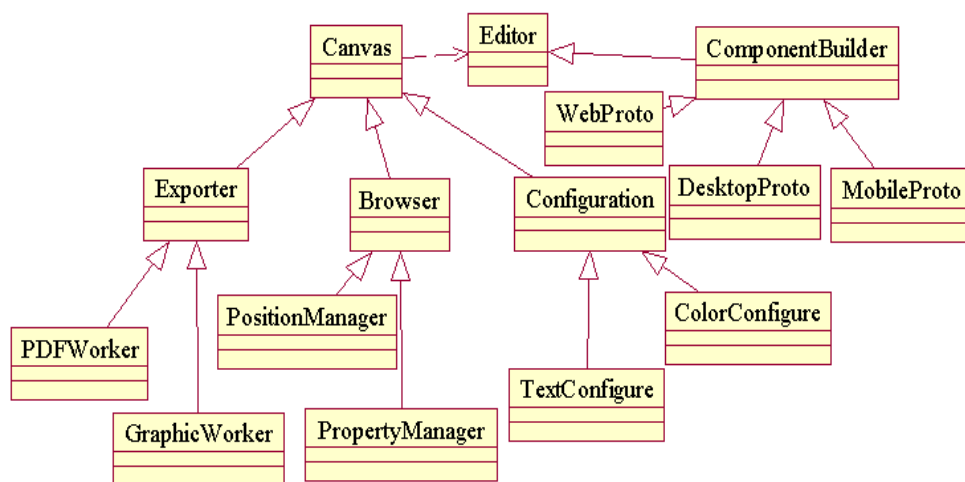


Рис. 3. Фрагмент діаграми класів системи

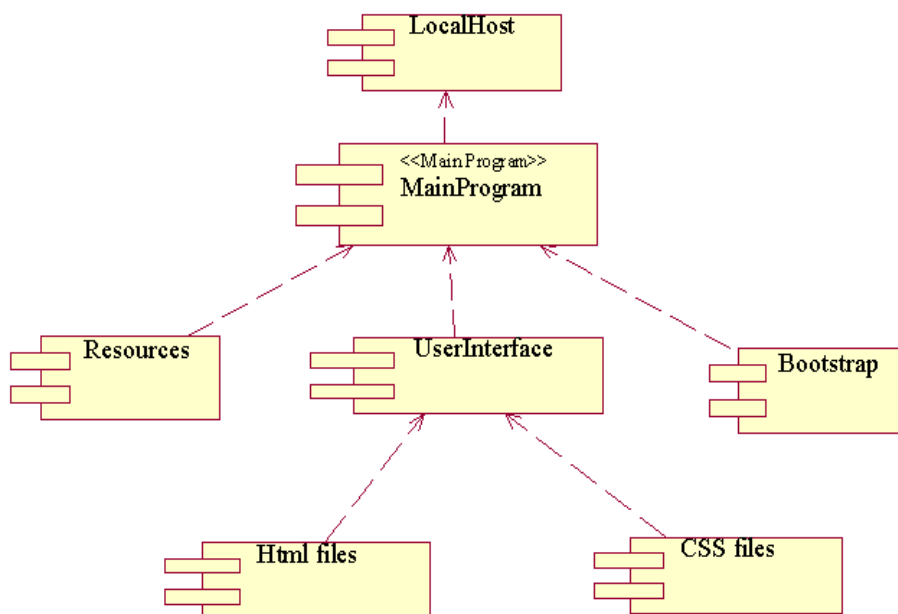


Рис. 4. Діаграма головних компонентів системи

Створені програмні файли розділені на 2 типу: контролери та подання. Основними методами розроблених контролерів є наступні:

- commonBehaviors, відповідає за загальну поведінку системи при її завантаженні;
- commonFunctions, реалізує функції відображення головного вікна системи;
- SVGTextLayout, імплементує функції підтримки і обробки векторної графіки;
- canvasHelper, відповідає за побудову базового контейнера для створення робочого полотна;
- canvasCareTaker, реалізує функції оновлення елементів в робочому просторі, де створюється прототип інтерфейсу.

Основними уявленнями є наступні:

- ApplicationPane, призначене для розмітки і розташування елементів головної робочої панелі;
- EditPageNoteDialog, виконує функції відображення вікна діалогу по модифікації налаштувань;
- ExportDialog, містить розмітку вікна обрання типу експорту створеного прототипу до файлу;
- PageDetailDialog, відповідає за зберігання елементів інтерфейсу вікна вибору типу контейнера;
- MainMenu, призначене для відображення елементів і складу головного меню системи.

**Висновки та перспективи подальших досліджень.** Розроблений проект кроссплатформеної розподіленої інформаційної системи прототипування зовнішнього вигляду програмних застосувань може бути використано різними групами користувачів для завдань швидкого створення концептуальних

інтерфейсів користувачів для різних цільових платформ, як то веб, десктоп чи мобільні програмні застосування. Завдяки ієрархічній структурі класів та використанню можливостей об'єктно-орієнтованого проектування стає можливим подальше наповнення проекту функціоналом шляхом гнучкого додавання нових чи розширення існуючих модулів системи. Можливими напрямками подальшого розвитку проекту є імплементація інших типів прототипів, підтримка додаткових засобів створення макетів та розширення компонентної бази окремих об'єктів інфографіки.

#### Список використаних джерел:

1. Слива М.В. Прототипування графічного інтерфейсу користувача як невід'ємна частина процесу розробки програмного забезпечення. *Вісник Нижневартівського державного університету*. 2013. № 1. С. 74–76.
2. Прототипування програми: від ідеї до робочого екрану. URL: [https://habr.com/company/mobile\\_dimension/blog/327452/](https://habr.com/company/mobile_dimension/blog/327452/).
3. Мардан А. Швидке Прототипування з JS. Гнучка Розробка на JavaScript. М.: ІТУС, 2014. 236 с.
4. Немтінов В.А. Віртуальне моделювання, прототипування і промисловий дизайн. Тамбов: ТДТУ, 2019. 339 с.
5. Поляков О.М. Основи швидкого прототипування. Оренбург: ОДУ, 2014. 128 с.
6. Глухова Л.А. Технології розробки програмного забезпечення. Мінськ: БДУІР, 2014. 97 с.
7. Варфел Т. Прототипування. М.: Манн, Іванов і Фербер, 2013. 389 с.
8. Вершиніна Е.В. Огляд моделей життєвого циклу розробки програмного забезпечення. Нижній Новгород: НДУ ім. Н.І. Лобачевського, 2010. 38 с.
9. 20 інструментів для прототипування: від швидкого і брудного wireframe до функціонального прототипу. URL: <https://medium.com/@denysergushki/>.
10. Николєнко О.І., Олейник П.П. Прототипирование и реализация графической формы заказа для информационной системы ресторанов быстрого питания. *Объектные системы*. 2015. № 10. С. 68–73.
11. Шибанов С.В., Пашкин А.А. Автоматизированное проектирование пользовательских интерфейсов. *Вестник Пензенского государственного университета*. 2016. № 16. С. 67–73.
12. Искра Н.А., Макоєд В.Н., Куница Е.Ю. Изучение и оценка подходов к разработке графического интерфейса пользователя. *Объектные системы*. 2015. № 10. С. 63–68.
13. Рудниченко Н.Д., Вычужанин В.В., Козлов А.Е. Модель front-end прототипа системы поддержки принятый решений мониторинга и управления рисками сложных технических систем. *Інформаційні управляючі системи та технології: Матеріали Міжнародної науково-практичної конференції (Іуст-Одеса-2015)*. 2015. С. 198–201.
14. Рудниченко Н.Д., Манькевич В.Н. Прототип програмного забезпечення експертних систем. *Інформатика, інформаційні системи та технології: тези доповідей п'ятнадцятої всеукраїнської конференції студентів і молодих науковців, м. Одеса, 27 квітня 2018 р.* Одеса, 2018. С. 71.
15. Rudnichenko N.D., Shibaeva N.O., Boyko V.D. Model prototype development of interactive interface of information system for monitoring ship technical system. *Інформаційні технології та комп'ютерна інженерія: матеріали статей п'ятої міжнародної науково-практичної конференції, м. Івано-Франківськ, 27-29 травня 2015 року*. Івано-Франківськ, 2015. С. 217–218.
16. Vychuzhanin V.V., Rudnichenko N.D., Shibaeva N.O., Boyko V.D. The development of user interface prototype of decision support system for risk management of complex technical systems. *Sustainability and Competitiveness in Business*. 2016. PP. 162–172.

#### References:

1. Sliva, M.V. (2013). Prototyping of the graphical user interface as an integral part of the software development process. *Visnik Nizhneartovskogo derzhavnogo universitetu – Herald of Nizhneartovsk State University*, 1, 74–76 [in Ukrainian].
2. Prototyping of the program: from the idea to the working screen. [Elektronij resurs]. [Prototyping of the program: from the idea to the working screen]. Rezhim dostupu: [https://habr.com/company/mobile\\_dimension/blog/327452/](https://habr.com/company/mobile_dimension/blog/327452/).
3. Mardan, A. (2014). *Shvidke Prototipuvannja z JS. Gnuchka Rozrobka na JavaScript* [Rapid Prototyping with JS. Flexible Development on JavaScript]. M.: ITUS [in Ukrainian].
4. Nemtinov, V.A. (2019). *Virtual'ne modeljuvannja, prototipuvannja i promislovij dizajn* [Virtual modeling, prototyping and industrial design]. Tambov: TDTU [in Ukrainian].
5. Poljakov, O.M. (2014). *Osnovi shvidkogo prototipuvannja* [Basics of rapid prototyping]. Orenburg: ODU [in Ukrainian].
6. Gluhova L.A. (2014). *Tehnologii rozrobki programnogo zabezpechennja* [Software development technologies]. Mins'k: BDUIR [in Ukrainian].
7. Varfel T. (2013). *Prototipuvannja* [Prototyping]. M: Mann, Ivanov i Ferber [in Ukrainian].
8. Vershinina, E.V. (2010). *Ogljad modelej zhitteвого ciklu rozrobki programnogo zabezpechennja* [Overview of software development lifecycle models]. Nizhnij Novgorod: NDU im. N.I. Lobachev'skogo [in Ukrainian].
9. 20 instrumentiv dlja prototipuvannja: vid shvidkogo i brudnogo wireframe do funkcional'nogo prototipu [20 tools for prototyping: from fast and dirty wireframe to a functional prototype]. Rezhim dostupu: <https://medium.com/@denysergushki/>.
10. Nikolenko, O.I., Olejnik P.P. (2015). Prototipirovanie i realizacija graficheskoi formy zakaza dlja informacionnoj sistemy restoranov bystrogo pitaniija [Prototyping and implementation of a graphic order form for the information system of fast food restaurants]. *Objektnye sistemy – Object systems*, 10, 68–73 [in Russian].

11. Shibanov, S.V., Pashkin, A.A. (2016). Avtomatizirovannoe proektirovanie pol'zovatel'skih interfejsov [Automated design of user interfaces]. *Vestnik Penzenskogo gosudarstvennogo universiteta – Herald of Penza State University*, 16, 67–73 [in Russian].

12. Iskra, N.A., Makoed, V.N., Kunica, E.Ju. (2015). Izuchenie i ocenka pohodov k razrabotke graficheskogo interfejsa pol'zovatelja [Study and evaluation of campaigns to develop a graphical user interface]. *Objektnye sistemy*, 10, 63–68 [in Russian].

13. Rudnichenko, N.D., Vychuzhanin, V.V., Kozlov, A.E. (2015). Model' front-end prototipa sistemy podderzhki prinjatij reshenij monitoringa i upravlenija riskami slozhnyh tehniceskikh system [The front-end model of the prototype of the decision support system for monitoring and risk management of complex technical systems]. *Materiali Mizhnarodnoi naukovopraktichnoi konferencii «Informacijni upravljajuchi sistemi ta tehnologii» – Proceedings of the International Scientific and Practical Conference “Information Control Systems and Technologies”*, 198–201 [in Russian].

14. Rudnichenko, N.D., Man'kevich, V.N. (2018). Prototip programmogo obespechenija proektirovanija jekspertnyh system [Prototype software for designing expert systems]. *Informatika, informacijni sistemi ta tehnologii: tezi dopovidej p'jatnadcjatoj vseukraïns'koï konferencii studentiv i molodih naukovciv. Odesa, 27 kvitnja 2018 – Informatics, information systems and technologies: abstracts of the reports of the fifteenth All-Ukrainian Conference of Students and Young Scientists. Odessa, April 27, 2018* [in Russian].

15. Rudnichenko, N.D., Shibaeva, N.O., Boyko, V.D. (2015). Model prototype development of interactive interface of information system for monitoring ship technical system. *Informacijni tehnologii ta komp'juterna inzhenerija: materialy statej p'jatoï mizhnarodnoi naukovopraktichnoi konferencii, m. Ivano-Frankivs'k, 27-29 travnja 2015 roku – Information Technologies and Computer Engineering: Proceedings of the Fifth International Scientific and Practical Conference, Ivano-Frankivsk, May 27-29, 217–218* [in English].

16. Vychuzhanin, V.V., Rudnichenko, N.D., Shibaeva, N.O., Boyko, V.D. (2016). The development of user interface prototype of decision support system for risk management of complex technical systems. *Sustainability and Competitiveness in Business*, 162–172 [in English].