

# Development of Mobile Application for Choreographic Productions Creation and Visualization

Antonii Rzheuskyi<sup>[0000-0001-8711-4163]</sup><sup>1</sup>, Aleksandr Gozhyj<sup>[0000-0002-2641-5947]</sup><sup>2</sup>,  
Andrii Stefanchuk<sup>3</sup>, Oksana Oborska<sup>4</sup>, Liliya Chyrun<sup>[0000-0003-4040-7588]</sup><sup>5</sup>,  
Olga Lozynska<sup>[0000-0002-5079-0544]</sup><sup>6</sup>, Khrystyna Mykich<sup>[0000-0002-4324-2080]</sup><sup>7</sup>,  
Taras Basyuk<sup>[0000-0003-0813-0785]</sup><sup>8</sup>

<sup>1,5-8</sup>Lviv Polytechnic National University, Lviv, Ukraine

<sup>2</sup>Petro Mohyla Black Sea National University, Nikolaev, Ukraine

<sup>3-4</sup>IT Step University, Lviv, Ukraine

antonii.v.rzheuskyi@lpnu.ua<sup>1</sup>, alex.gozhyj@gmail.com<sup>2</sup>,

andy.stef.2512@gmail.com<sup>3</sup>, oksana949@gmail.com<sup>4</sup>,

lchirun21@gmail.com<sup>5</sup>, Olha.V.Lozynska@lpnu.ua<sup>6</sup>,

khrystyna.i.mykich@lpnu.ua<sup>7</sup>, Taras.M.Basyuk@lpnu.ua<sup>8</sup>

**Abstract.** The task of the work is to develop a mobile application for creating and visualizing choreographic productions. The application is implemented for phones and tablets running under the operating system iOS. During the performance of the work, the process of presenting choreographic productions was examined and the problems encountered by choreographers in this case were identified. Specific cases are considered, which clog and complicate the work of the choreographer. After analyzing existing analogues programs, their shortcomings and requirements are formed which will cover all problems of users. The application was developed using the Swift programming language in the xCode environment. The role of the database is Realm. The architecture of the system is designed according to the standards of the MVC template offered by Apple. The result of the work has been successfully tested and prepared for release on the market. The developed application allows you to visualize choreographic drawings, view animated transitions between them and save information about all participants of the statement. In addition, the application has a functional that is not found in any of the analogues: music support, configuration of scene sizes, color change of dancers and pattern patterns. Convenience of the application and additional features make the application relevant for both choreographers and performers.

**Keywords:** Information System, Mobile Application, Choreographic productions.

## 1 Introduction

Today, mobile devices are very popular and do not stop at getting popular. It will soon be difficult to imagine a person without a smart device. Such a crazy popularity is due to the fact that a person in the modern world should always be accessible and in

touch. Twenty years ago, the mobile phone was used only as a communication medium and had a slightly wider functionality than a simple phone. Subsequently, there was a spare book and some tools for exchanging information. Now, with the help of a medium-priced mobile phone, you can use the internet, make high-quality photos, shoot videos, play games that can have better graphics than some computers [1-5]. Due to its compactness and wide-ranging functionality, the mobile device can easily replace a computer or laptop for many people in everyday life [6-9].

Together with the growing popularity of mobile gadgets, it is gaining momentum and developing mobile applications that can be applied in completely different spheres of human activity: business, education, entertainment, finance, food, health, medicine, music, navigation, news, sports, travel, and communication [10-16]. Most applications are designed to simplify certain processes in the listed areas of activity [17-21]. It does not matter whether it is a daily reminder to drink in the morning water, or a taxi call, by pressing only one button [22-29]. A mobile application can be a convenient substitute for everyday affairs, such as entering a repository, and used for complex processes such as accounting for finances in the bank [30-37].

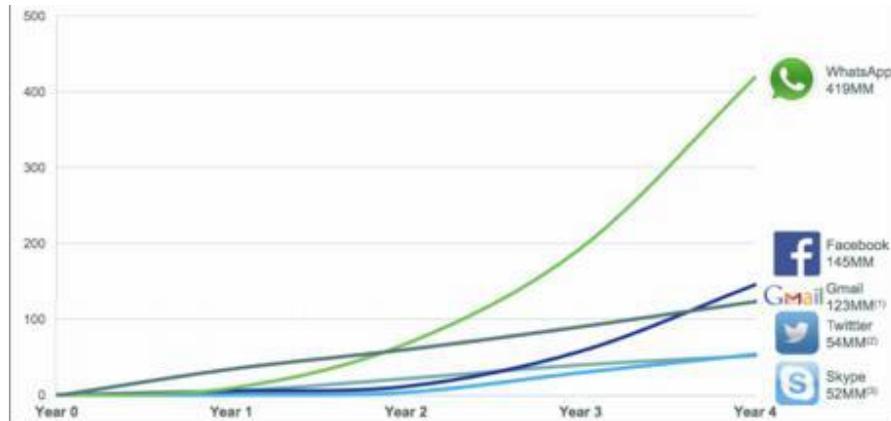
The aim of the work is to develop an application that will facilitate the work of choreographers, who often deal with the production of dance numbers, and reduce the time it takes to create a presentation of productions. The application is intended to solve the problem of the inconvenience of drawing dance statement on paper. With a large number of participants in the room, the drawing process becomes complicated and inconvenient. When changes occur in the production of a choreographer, it is necessary to repaint and correct many drawings. Changing members of the number also causes additional problems.

Using a mobile application, the user can easily visualize their staging regardless of the number of participants and the size of the scene [38-41]. You will also be able to view all the transitions and tie the particular members of the number to their names.

The application will find its use among choreographers in dance studios and schools. It will also be useful to dancers and roommates in terms of perception of the entire production through a visual presentation. Also, the application will help you to view and remember your position in the room.

## **2 The Process of Creation of Choreographic Responses and the Use of Mobile Applications for Their Representation**

The rapid development of mobile devices in recent years has begun a new trend in the field of software development. And now the global mobile application market is growing at a fast pace and is one of the most popular. For example, in 2013, the Instagram application was redeemed by Facebook for almost \$ 1 billion. After that, Facebook bought WhatsApp, but already for \$ 19 billion. [6] Information on the number of active users who use these products every month on the market for more than four years is shown in Fig. 1. You can notice that for 4 years of WhatsApp, its monthly audience reaches the figure of 419 million users [7].



**Fig. 1.** Number of user accounts active every month

For many people, a mobile phone or tablet is completely capable of replacing a laptop or a stationary computer at the expense of its power, mobility and a wide range of functionalities [8-14]. Accordingly, users need a wide range of different software - from office applications and applications to solve everyday business tasks to entertaining applications, games, and more. New programs are an essential element in the development of commercial, industrial and other spheres of human activity [15-21].

Most companies, besides creating a web page for product presentation or service delivery, also create a mobile application that sometimes has a wider functionality (for example, the Instagram web page gives you the opportunity to just comment and view other people's posts, while the mobile application allows posting their entries to the social network) or a mobile version of a site that is no different from the original web page, only optimized for devices with smaller screen sizes [6-9].

In general, mobile applications can be divided into three types [22-28]:

1. Web Applications, Sites. The modern mobile device provides a similar functionality when working on the Internet, like a regular computer or laptop, thanks to the support of the HTML 5 technology. In addition, it is one of the best options for fast start with minimal cost and good functionality. Mobile sites are a universal solution, since browsers are available for each operating system and adhere to the same unified standards. The page layout may vary in different browsers, depending on the kernel on which they are built, but these are minor points that are detected and corrected during the development process.
2. Native software. The development is conducted in "native" programming languages, which allows you to take advantage of all the features and benefits of the OS and provides full access to resources that are open to the programmer. That is why this type of software is the most functional and shows the greatest productivity. This kind of software depends on the platform of the platform directly.
3. Hybrid software - guarantees access to all functional mobile devices. It is a combination of native software with third-party markup languages. Development is done using various frameworks, CSS, HTML, etc. This type of application is a good so-

lution for those who want to use modern web technologies along with the benefits of native software [8].

Choreography is the art of dancing as a sequence of steps, motions, figures to create the best stage effect. The art of creating stage dance is based on the manipulation of abstract elements of the movements of the human body: space, form, time, energy within the context of an emotional context in order to express a unique creative voice. The language of choreography moves is the language of ballet dance technique, modern or jazz dance, hip hop, folk dance, ritual dance, or routine everyday movements. Choreography is trying to achieve in a dance composition of organic integrity, rhythmic and nerimetric articulation, themes and variations, repetition and improvisation. The term choreography is also used to articulate performances in certain sports: artistic and sports gymnastics, figure skating, synchronized swimming [9]. In recent years, interest in contemporary dance has grown dramatically in Ukraine and in general around the world - the number of dance centers and studios, amateur and professional groups has increased. More and more people want to express themselves in dance. One can distinguish the following cases where the concept of choreographic statement applies [29-37]:

- Production of choreography for clips and commercials;
- Preparation of the artist for filming the music video;
- massive flash mobs;
- performance of the show performance of the dance team / group;
- statement of support groups;
- Production of choreographic accompaniment for the performance of the artist. [10]

A good dance setting is an easy task and requires experience, musicality and certain skills, including improvisation, artistic thinking and the ability to express your idea in dance. First, the director determines the idea of his room, inspired by some source. Then he selects the musical accompaniment, which will best emphasize the chosen idea. In order to make the dance the most exciting, voluminous and use the entire stage space, it is necessary to correctly arrange the drawings, which also requires imagination, experience and strongly depends on the number of people in the group. Well-chosen pictures can perfectly complement music, choreographic movements and much better to convey the idea of the number. The spectacular and distinctive change in the arrangement of dancers is one of the main techniques that can keep the viewer in tension and focus on the required areas of the scene.

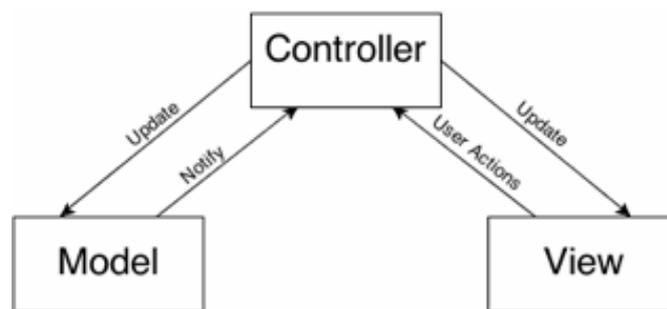
### **3 Development of Mobile Application for Creation and Visualization of the Choreographic Statements**

As the main architectural template we decided to use the MVC template (model - a look - the controller) as recommend to Apple this approach when developing appliques and it is not heavy in realization [11] especially, at observance of the basic rules of MVC types it is possible repeatedly perevikoristovuvat in many places.

MVC is one of Cocoa main units of a framework and undoubtedly is most often used by developers in general. It divides objects depending on their role in the application and the zaklta e to divide the code which also depends on roles [32-41]:

- the model is an object which contains certain data the appendix describes business logic and defines how they can be operated. For an example, in a rozroblyuvany zastosun the classes Choreography, Scene and others are models.
- a look is an object which is responsible for visual representation of data of model and elements with which the user can interact. Usually, all this objects, which nasl\_duyutsya from a class to an uiviye.
- the controller is synchronizes all work. It has access to data from model and submits them by means of types, listens to events, changes and directs to data when there is a need.

Normal realization of this template design means that each object belongs to one of these groups. Communication between a look and model by means of the controller can be presented by the drawing.



**Fig. 2.** Communication in the MVC template.

The model notifies the controller of changes in its data, and the controller responds with updates in the form. The view can then notify the controller of the actions being taken by the user and either update or update the model as needed, or obtain the data necessary for the user.

All this separation is precisely for the delimitation of the code and the possibility of its reuse. Ideally, the look must be completely isolated from the model. If it does not depend on the particular implementation of the model, it can be easily reused with another model for the presentation of completely different data. MVC on iOS.

To begin with, make sure that each class in the project is either a controller, or a look, or a model. You should not combine the responsibility of two different roles into one class. Next, you need to create three groups of projects to group the code, each for the appropriate category.

One of the benefits of an MVC template is the ability to design parts project independently of each other, and then easily combine them. The diploma project will be divided into several components:

- lists that will be responsible for displaying and filtering choreographies;
- a scene that will contain all the logic of displaying dancers, their transitions and manipulations;
- audio is a separate component for separating all work with audio files.

The model notifies the controller of changes in its data, and the controller responds with updates in the form. The view can then notify the controller of the actions being taken by the user and either update or update the model as needed, or obtain the data necessary for the user.

All this separation is precisely for the delimitation of the code and the possibility of its reuse. Ideally, the look must be completely isolated from the model. If it does not depend on the particular implementation of the model, it can be easily reused with another model for the presentation of completely different data. MVC on iOS.

To begin with, make sure that each class in the project is either a controller, or a look, or a model. You should not combine the responsibility of two different roles into one class. Next, you need to create three groups of projects to group the code, each for the appropriate category.

One of the benefits of an MVC template is the ability to design parts project independently of each other, and then easily combine them. The diploma project will be divided into several components:

- lists that will be responsible for displaying and filtering choreographies;
- a scene that will contain all the logic of displaying dancers, their transitions and manipulations;
- audio is a separate component for separating all work with audio files.

**Using Design Templates.** Design patterns are template approaches to solving certain common problems that often arise when designing architectural solutions. The advantage of patterns is that they provide a common understanding of the problem and its solution for developers. This means that developers can write in completely different languages, but understand how to solve a problem with a certain template. Also, patterns can be built the project is more structured and clear. Therefore, when designing a project structure, I decided to use the following design patterns:

**Observer.** In the design pattern of Observer (Observer), one object notifies its subscribers of changes in their status. Engaged objects should not know about each other - thus providing a distinct design. This template is most often used to alert certain objects about changing properties.

When using the MVC pattern, you must allow the model to communicate with the look, but without direct links between them. And here the template is useful.

**Notifications** is often confused with push and local alerts. Notifications are based on the model "Subscribe-and-publish", which allows an object (publisher) to send alerts to its subscribers / listeners. Publisher should never know any information about their subscribers.

**KVO** is an object may ask to be notified of a change either of its own or of any other property [12].

For the project, this template will be used to observe changes to objects in the database and update the interface according to new values.

**Single (singleton).** The Singleton Design Pattern (single) guarantees that the class will only exist in one instance from a common point of access to it. Usually "lazy boot" is used to create a single one when it was created for the first time. Apple uses this template very often. For example: `NSUserDefaults.standardUserDefaults()`, `NSUserDefaults.standardUserDefaults()`, `UIFileManager defaultManager()`, `UIScreen mainScreen()`, `UIApplication sharedApplication()`, all return the object to the same. There are cases in which it makes sense to have exactly one instance of a class. For example, there is only one instance of the application and one main screen for the device, so you only need one for each of them; the class that is responsible for configuring the application – it's easier to implement stream-safe access to one global resource (such as a configuration file) than having many classes that can edit it at a time. Within the project, it makes sense to create only one instance of the class to work with the database.

**Decorator.** Designer template "decorator" allows you to dynamically add functionality to an object without changing its base code. It is an alternative to creating subclasses that change class behavior by wrapping it with another object. Swift has two very common implementations of this template: Extensions and Delegation.

Expansion is an extremely powerful mechanism that allows you to add a new functionality to already existing classes, listed types, structures without the need to create subclasses. A huge advantage is that you can "expand" the code to which the programmer has no access and extend its operation without knowledge of the implementation. For example, new methods that were added during compilation can be performed as standard methods of the extended class. This is slightly different from the classical definition of the decorator, because the extension does not contain an instance of the class whose function it extends. Classes, of course, can redefine the methods of their father, but the expansion does not allow it. The methods and properties of the extension may not have the same name as the methods and properties of the original class.

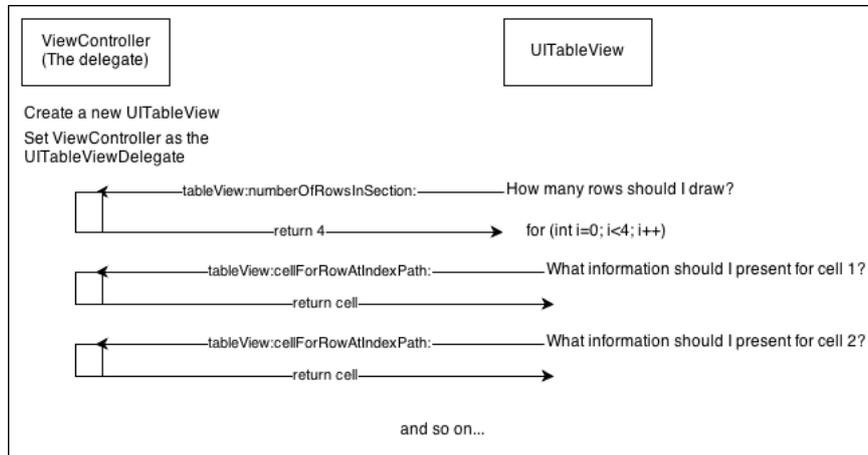
Delegation is a mechanism in which one object acts on behalf of another object.

For example, when we use `UITableView`, one of the methods we are forced to implement is `tableView(_:numberOfRowsInSection:)`.

It is impossible to predict for `UITableView` how many rows each section will have, as it depends on the application. Thus, the task of calculating the number of rows in each section is passed to the delegate `UITableViewDelegate`, which allows the class to be independent of the type of data it displays. In Fig. 3. an explanation of the actions that occur after the creation of the new `UITableView` is shown.

`UITableView` displays the table. However, the class needs some information that it does not have. Then he addresses the delegates, sends them a message asking for additional information. In Objective-C.

The implementation of this template allows the class to determine obligatory and optional methods through protocols. This is a very important design template and Apple uses it in most `UIKit` classes: `UITableView`, `UITextView`, `UITextField`, `UIPickerView`, `UICollectionView`, etc.



**Fig. 3.** Messaging between UITableView and the delegate controller

**Keeper (Memento).** The "Memento" design template is used when there is a need to save and restore something. This can be to save the state of the game or save the settings in the program. Within the framework of the project, it will be possible to cancel the actions made by the user. To do this, each user action will be recorded on the event stack and, if necessary, canceled, the last action will be thrown out of the stack and the opposite action is applied.

**Designing a database.** As a result of the work carried out and described in the previous section, a mobile Realm database was selected. This is a non-relational database that allows you to work hard with objects stored on a disk, as with the usual objects in the program. Given the specifics of the software being developed, the database structure will not be complicated and will not contain a large number of links. To meet the needs of the user, only the combination of basic properties will be sufficient to preserve the choreographic drawings that the user creates. The main things will be choreography, a template, a dancer, a picture, a scene configuration, a position and a musical accompaniment. Choreography - stores information about the setting, all the drawings, from which it is composed, their order and transitions between them, the dancers, the musical track and the configuration of the scene. Contains the following fields:

- • name - the name of the staging;
- • date of creation;
- • date when recent changes were made;
- • scene configuration;
- • Id - a unique identifier for the distribution of choreography in distribution;
- • dancers - a list of all dancers participating in the production;
- • drawings - a list of all created drawings.

Template - saves information about one specific drawing without being tied to specific dancers or music. Contains the following fields:

- template name;
- Positions is a list of positions forming a picture.

Dancer - stores information about the dancer in the production. Contains the following fields:

- name identifies the dancer; may be empty, which is not essential for the choreographer;
- color is color to display the dancer on the stage.

Scene Correction - saves information about scene sizes. Contains the following fields:

- scene width (number of vertical lines);
- length of the scene (number of horizontal lines);
- the size of the field on the sides of the stage;
- a flag to display or hide space outside the scene.

All these sizes are given in relative units, which will be displayed as one square of the scene. The scene itself is essentially a set of lines that form a rectangle and allow you to place dancers at the intersection of these lines.

Drawing - saves information about the arrangement of dancers on the stage. Contains the following fields:

- positions is a list of all positions where dancers are placed;
- the duration of the transition to the next picture is given in seconds;
- name part of the setting, which includes the picture.

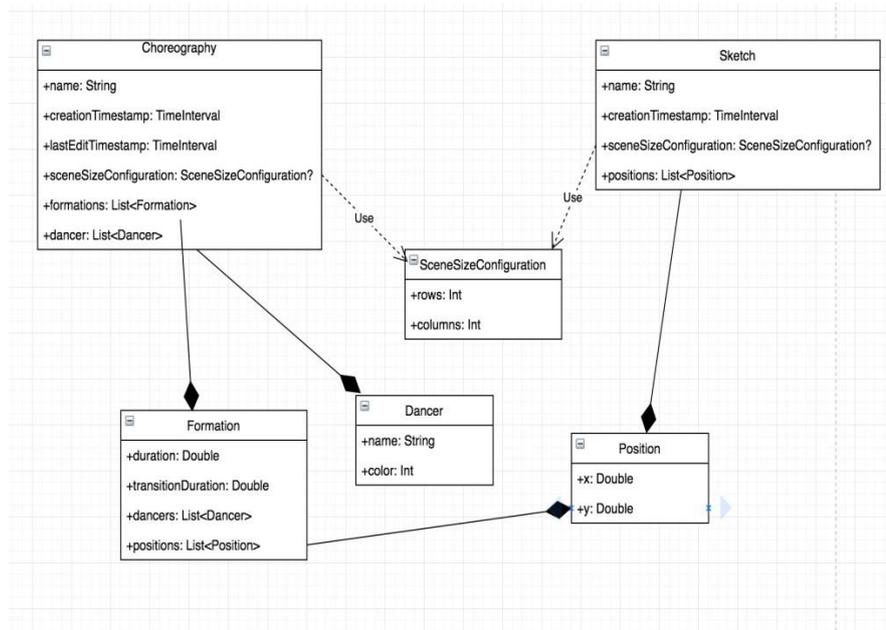
Position -s stores information about placing a specific dance on the stage. Contains the following fields:

- x and y are the relative coordinates of the placement.
- musical accompaniment - saves the information needed to synchronize pictures with music. Contains the following fields:
- a list of time frames that indicate the beginning and end of a particular picture or transition;
- the path to which the audio file will be stored.

All of the above entities will be presented as descendants of the base Object class, which is essential when working with Realm and provides a link between the objects in the program and the stored data on the disk. Object allows you to define primary keys, indexed properties, ignored properties. At the same time, all these classes form the structure of the database - each class is a separate table with the corresponding name and fields. The schematic of the database is shown in the figure.

The main users of the application will be choreographers of various dance schools and studios, choreographers, who create stage numbers for singers and support group trainers. Each of these users will create posts from a certain number of participants, so it's worthwhile to allocate two groups of users:

- a choreographer whose task is to visualize his production;
- stage participant, for which you need to know all positions during production.



**Fig. 4.** Database schema

Choreographer, having created his statement, will be able to share it with his students or participants of the issue. This will be done through a deep link that the choreographer will generate and send to others. The link will contain all information about the created choreography in the binary representation. By going through this link, the user will be copied this data and he will be able to view the setting on his device.

The user enters the application, goes to the choreography list and chooses what he wants to do: edit the already created choreography or create a new one. When creating a new choreography, you must specify the name of the setting and configure the size of the scene. After that the user will be presented with a choreography editing window with the first empty picture. On this screen, the user will be able to perform the following operations:

- Add new dancers to the scene in the form of points by touching the scene where the dancer should stand;
- change the position of the added dancers by dragging them to a new position;
- set or change the color and name of the dancers;
- set one of the modes of drawing to the lines of the scene (to the intersection of lines or to the middle of a particular line);
- create a new drawing, which will first be a copy of the last drawing, and then the transition between the drawings will be set in the way of dragging dancers into their new positions;
- switch between the pictures, which will be accompanied by an animated transition of all dancers from their old positions to new ones;

- view the entire choreography with a certain speed.
- open the choreographic detail screen.

On the details screen, we can perform the following operations:

- delete a specific drawing;
- insert an intermediate drawing;
- Change the configuration of the scene;
- Generation of the link for the distribution of the created choreography.

All functionalities described above will be accessible to each user. To monetize applications, certain restrictions will be introduced that can be removed by purchasing a premium version of the application. Along with removing restrictions, the user will have an additional functionality available. The limitation will be that the average user will be able to keep only 5 choreographies saved. After reaching the limit, the user will need to remove one of the saved choreographies or purchase a premium version. Also, premium users will be able to add music to choreography.

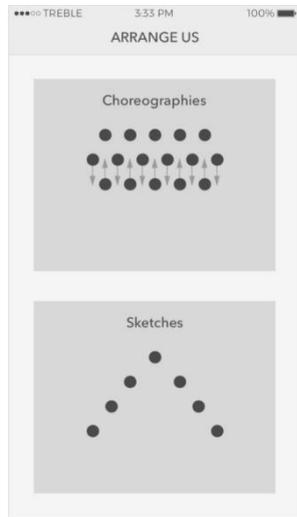
To work, the user interacts with the system, which in turn generates a query to the database. After receiving the query results, the database sends a response to the system that formats the data and displays it to the user.

## 4 Prototypes of Interfaces

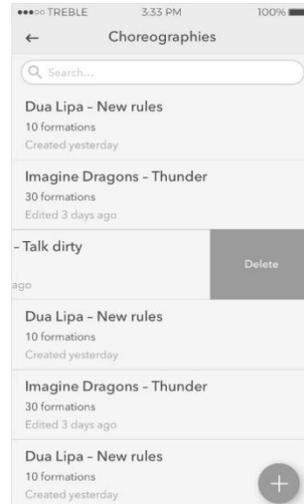
Based on a certain functionality We have created a basic schema of application screens and transitions between them. So, the main screens are:

- Main menu
- Choreography list
- List of templates
- New choreography / template creation screen
- Scene configuration screen
- Choreography editing screen
- Choreography screen
- Screen with details of choreography

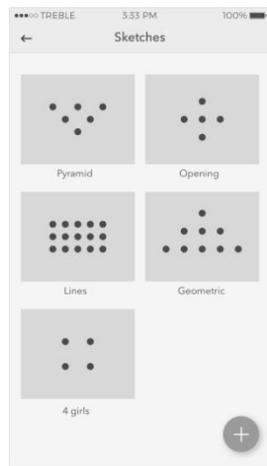
The next step was to plan the application navigator and cross-screen navigation. The home screen will contain buttons to go to the list of all choreographies or templates. These lists will have the opportunity to go to the new choreography / template screen or go directly to the choreography / template edit screen. On the screen editing will be available all the functionality that is required for visualization of the productions, in particular the transition to the viewing mode. It will also be possible to go to the screen with details for quick editing and access to specific drawings. To create the prototype, an online service for inVision prototyping was used. Prototypes of the above screens are shown in (Fig 5–10).



**Fig. 5.** Main menu.



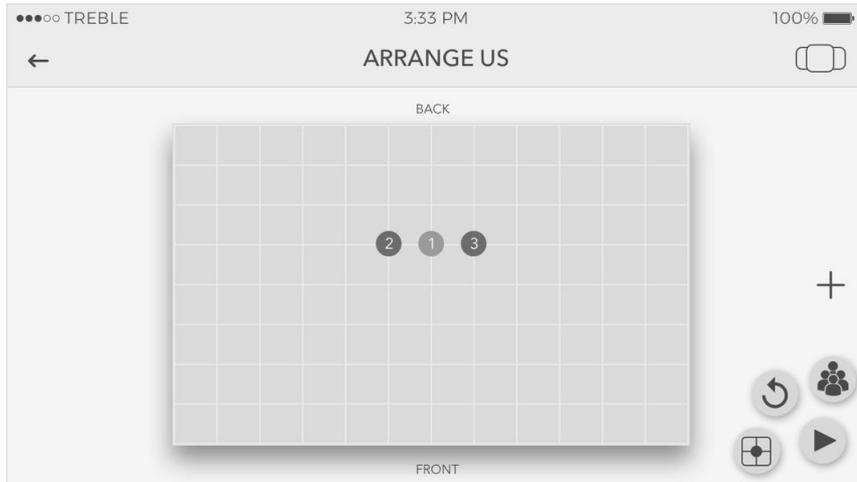
**Fig. 6.** List of choreographies.



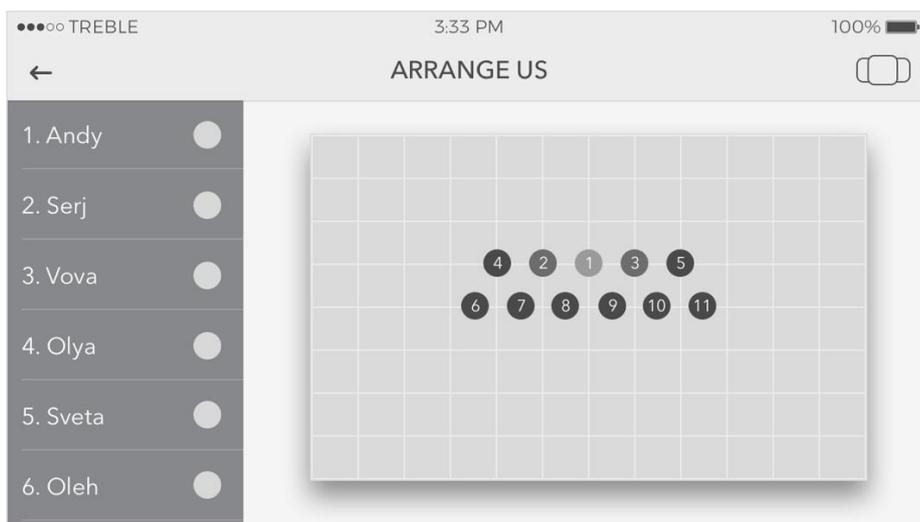
**Fig. 7.** List of templates



**Fig. 8.** Configuration of the scene



**Fig. 9.** View of scene



**Fig. 10.** Editing dancers.

## **5 Program Realization of Mobile Applications for Creation And Visualization of Choreographic Responses**

When designing applications, many classes have been created that can be broken into several groups: data structures, work with a local database, auxiliary general classes, constants and description of application styles, classes for displaying graphical information, class controllers, classes for the operation of the scene. Each class of the model described in the design section is a descendant of the base class of the Realm-

Object database. Object allows you to work with data recorded on a disk, as with simple objects. For unified operation with all objects of the database a general manager was created - the RealmService class. This is a wrapper, through which there is an appeal to the database. If you need to switch to another database, you will simply need to redefine the logic of all methods in this class.

Template methods allow us to apply basic operations to any type that are descendants of the Object. These methods include the following:

- add () - create / update row / rows in a table;
- update () - updates the specific table row properties;
- delete () - delete the line;
- removeAll () - delete the table;
- write () - write any changes to the table, changing the properties of the objects.

Another responsibility of this class is the initial configuration of the database and the migration between different versions of the database schema.

The specificity of the Realm database is such that it is enough to only get once the objects of a certain type that will be stored in the Results structure <object type>, after which all changes, addition or deletions will be made and this structure will be actual in any -any moment of time.

Given this feature, the ChoreographiesProvider class was created to access choreographed manipulation from any location in the program. The RealmService class described above is used for manipulation. Also, this class provides the opportunity to get a list of all choreographies and sort this list by a certain property.

Among the available choreographed operations are the following methods: adding a new drawing, adding a new dancer, changing the dancer's position in a particular drawing, changing the name and color of the dancer, removing the picture, and inserting an interim picture.

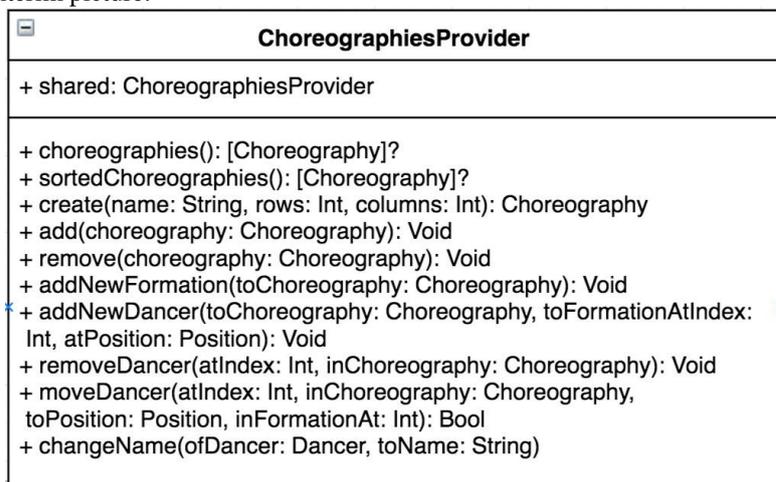


Fig. 11. Class chart for choreography

The central element that implements the business logic of applications is the scene used to display choreography, dancers and all the transitions between drawings, and for the direct creation of productions. The logic of the scene was divided into separate components, namely:

- visual display;
- component to calculate the size of the scene;
- component for placing dancers on the stage;
- component for dragging the dancer on the scene.

To display the scene, the descendant of the UIView class - SceneView was created and the drawing logic was implemented. The input gives the configuration of the scene size and screen size of the device. The result is a path containing a set of horizontal and vertical lines that are placed with the same intralus from each other, forming a certain number of squares that correspond to the desired size of the scene.

To place dancers on the stage, a separate class -SceneUnitLocator, which receives a touch point with a scene that is relative to the screen size of the device, has been implemented. This point is converted to the coordinates of the scene, after which the nearest point where you can place the dancer is calculated. There are 2 options for calculating the closest point on the stage: the line intersection or the middle of the line. If the point is valid and is within the scene, then a new dancer is added in the form of a circular point.

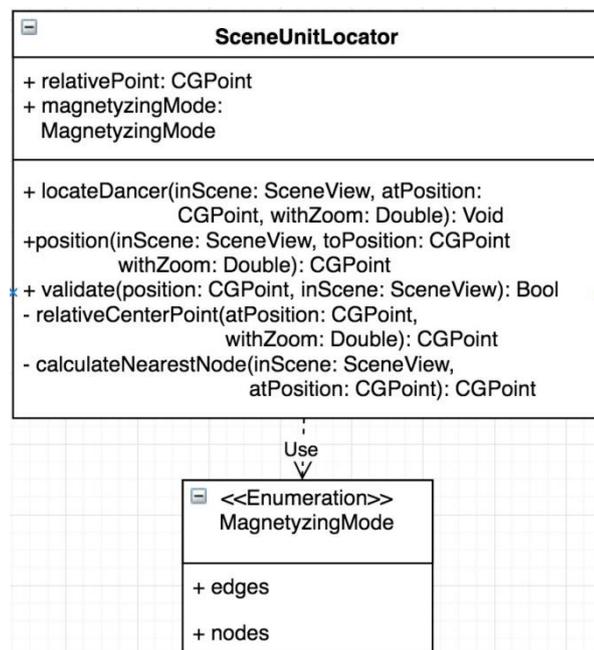
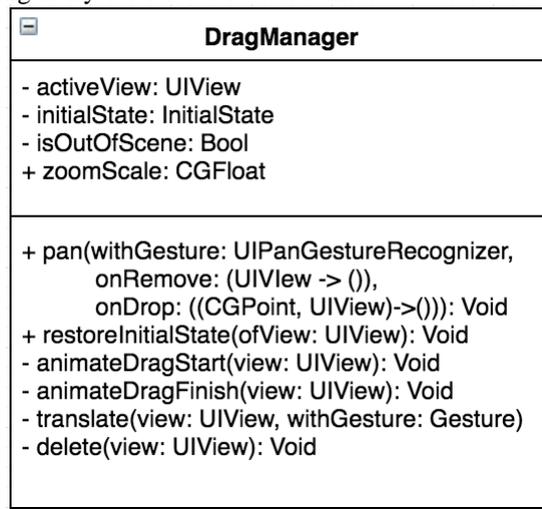


Fig. 12. A class diagram for placing dancers on stage.

To change the dancer's position, the DragManager class was implemented on the stage. After touching the dancer, his performance is animated by increasing. During the dragging of the finger position of the user, the dancer also changes his position accordingly. In case of going out of the scene, the user is asked to remove the selected dancer from the staging. If a user releases a dancer within a scene, then the nearest point is calculated according to the same logic as when adding a dancer. This class supports dragging many dancers at the same time.



**Fig. 13.** A class diagram for dragging dancers on stage

There is a class in which there is all the logic of drawing the scene and the dancers on it, but the size of the grid in this case is much smaller because the generated image is used only as a preview.

## 6 Conclusions

During the work, a mobile application under the iOS platform was developed to create and visualize choreographic productions, which greatly facilitates the work of the choreographers and helps the participants of the productions. The main purpose of the application is to create a visual representation of the movement of dancers on the stage. The following work was carried out to accomplish the task: The process of presenting choreographic productions is considered. It has been found that this process often causes certain problems in choreographers, since there is no unambiguous and obvious way of visualizing the movement dancers. The imperfection and inconvenience of making staging on paper is proved.

1. Analysis of existing software products on the market, helping to solve the problem of visual presentation of productions. Their drawbacks were discovered and it was concluded that it is possible to create a better and better product.

2. A detailed study was conducted among potential users who encountered the problem of visualizing choreography. As a result of the research, the relevance of software development has been confirmed and the task and requirements of users in the system to be developed are clearly formed.
3. Selected development tools, including xCode development environment, Swift programming language, Realm database, and other minor tools.
4. A system, the main architecture of which is MVC, is designed; the main design patterns are selected; The database and basic prototypes of the user interface are designed.
5. Successful modular testing and testing of the user interface after product implementation has been carried out.

## References

1. Kazarian, A., Holoshchuk, R., Kunanets, N., Shestakevych, T., Rzhеuskyi A.: Information support of scientific researches of virtual communities on the Platform of Cloud Services. *Advances in Intelligent Systems and Computing III*, vol.871. pp. 301-311 (2018)
2. Rzhеuskyi, A., Kunanets, N., Stakhiv, M.: Recommendation System "Virtual Reference". In: 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT), vol. 1, pp. 203-206 (2018).
3. Lypak, H., Rzhеuskyi, A., Kunanets, N., Pasichnyk, V.: Formation of a consolidated information resource by means of cloud technologies. In: 2018 International Scientific-Practical Conference on Problems of Infocommunications Science and Technology, PIC S and T, Kharkiv, pp. 157-160 (2018).
4. Rzhеuskyi, A., Kunanets, N., Kut V.: The analysis of the United States of America universities library information services with benchmarking and pairwise comparisons methods In: 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2017, vol.1, pp. 417-420 (2017).
5. Kaminskyi, R., Kunanets, N., Rzhеuskyi, A.: Mathematical support for statistical research based on informational technologies. In: 14th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer, vol. 2105, pp. 449-452 (2018).
6. What is the mobile application market in Ukraine? Access mode: <http://goo.gl/MTpr9w>
7. What's up, WhatsApp? Facebook's \$ 19B baby hit by outage, <https://goo.gl/Iatt7l>
8. Development of web applications, mobile applications and portals, <http://goo.gl/MmFGUH>
9. What is choreography? <https://uk.wikipedia.org/wiki/Choreography>
10. Choreographic statement as a service, <https://mywaydance.com/shop/uslugi/postanovki/>
11. Model-View-Controller (MVC) in iOS: A Modern Approach [Accessibility]: Accessibility: <https://developer.apple.com/library/content/documentation/General/Conceptual/CocoaEncyclopedia/Model-View-Controller/Model-View-Controller.html>
12. Introducing iOS Design Patterns in Swift - Part 1/2, <https://goo.gl/s5HjdR>
13. Shakhovska, N., Vysotska, V., Chyrun, L.: Features of E-Learning Realization Using Virtual Research Laboratory. In: Computer Science and Information Technologies, Proc. of the XI-th Int. Conf. CSIT'2016, 143–148 (2016)

14. Naum, O., Chyrun, L., Kanishcheva, O., Vysotska, V.: Intellectual System Design for Content Formation. In: Computer Science and Information Technologies, Proc. of the Int. Conf. CSIT, 131-138 (2017)
15. Vysotska, V., Chyrun, L., Chyrun, L.: Information Technology of Processing Information Resources in Electronic Content Commerce Systems. In: Computer Science and Information Technologies, CSIT'2016, 212-222 (2016)
16. Vysotska, V., Rishnyak, I., Chyrun, L.: Analysis and evaluation of risks in electronic commerce, CAD Systems in Microelectronics, 9th International Conference, 332-333. (2007)
17. Lytvyn, V., Vysotska, V., Burov, Y., Veres, O., Rishnyak, I.: The Contextual Search Method Based on Domain Thesaurus. In: Advances in Intelligent Systems and Computing, 689, 310-319 (2018)
18. Lytvyn, V., Vysotska, V., Veres, O., Rishnyak, I., Rishnyak, H.: Classification methods of text documents using ontology based approach, Advances in Intelligent Systems and Computing, 512, 229-240. (2017)
19. Lytvyn, V., Vysotska, V., Chyrun, L., Chyrun, L.: Distance Learning Method for Modern Youth Promotion and Involvement in Independent Scientific Researches. In: Proc. of the IEEE First Int. Conf. on Data Stream Mining & Processing (DSMP), 269-274 (2016)
20. Shakhovska, N., Vysotska, V., Chyrun, L. Intelligent Systems Design of Distance Learning Realization for Modern Youth Promotion and Involvement in Independent Scientific Researches. In: Advances in Intelligent Systems and Computing 512. Springer International Publishing AG, 175-198 (2017)
21. Vysotska, V., Chyrun, L., Chyrun, L.: The Commercial Content Digest Formation and Distributional Process. In: Computer Science and Information Technologies, Proc. of the XI-th Int. Conf. CSIT'2016, 186-189 (2016)
22. Lytvyn, V., Vysotska, V.: Designing architecture of electronic content commerce system. In: Computer Science and Information Technologies, Proc. of the X-th Int. Conf. CSIT'2015, 115-119 (2015)
23. Kanishcheva, O., Vysotska, V., Chyrun, L., Gozhyj, A.: Method of Integration and Content Management of the Information Resources Network. In: Advances in Intelligent Systems and Computing, 689, Springer, 204-216 (2018)
24. Vysotska, V., Chyrun, L.: Analysis features of information resources processing. In: Computer Science and Information Technologies, Proc. of the Int. Conf. CSIT, 124-128 (2015)
25. Lytvyn, V., Vysotska, V., Peleshchak, I., Rishnyak, I., Peleshchak, R.: Time Dependence of the Output Signal Morphology for Nonlinear Oscillator Neuron Based on Van der Pol Model. In: International Journal of Intelligent Systems and Applications, 10, 8-17 (2018)
26. Lytvyn, V., Vysotska, V., Veres, O., Rishnyak, I., Rishnyak, H.: The Risk Management Modelling in Multi Project Environment.. In: Computer Science and Information Technologies, Proc. of the Int. Conf. CSIT, 32-35 (2017)
27. Lytvyn, V., Vysotska, V., Pukach, P., Bobyk, I., Pakholok, B.: A method for constructing recruitment rules based on the analysis of a specialist's competences. In: Eastern-European Journal of Enterprise Technologies, 6/2(84), 1-14 (2016)
28. Vysotska, V., Hasko, R., Kuchkovskiy, V.: Process analysis in electronic content commerce system. In: Proceedings of the International Conference on Computer Sciences and Information Technologies, CSIT 2015, 120-123 (2015)
29. Lytvyn, V., Vysotska, V., Pukach, P., Vovk, M., Ugryn, D.: Method of functioning of intelligent agents, designed to solve action planning problems based on ontological approach. In: Eastern-European Journal of Enterprise Technologies, 3/2(87), 11-17 (2017)
30. Vysotska, V., Fernandes, V.B., Emmerich, M.: Web content support method in electronic business systems. In: CEUR Workshop Proceedings, Vol-2136, 20-41 (2018)

31. Lytvyn, V., Sharonova, N., Hamon, T., Vysotska, V., Grabar, N., Kowalska-Styczen, A.: Computational linguistics and intelligent systems. In: CEUR Workshop Proceedings, Vol-2136 (2018)
32. Gozhyj, A., Vysotska, V., Yevseyeva, I., Kalinina, I., Gozhyj, V.: Web Resources Management Method Based on Intelligent Technologies, *Advances in Intelligent Systems and Computing*, 871, 206-221 (2019)
33. Su, J., Sachenko, A., Lytvyn, V., Vysotska, V., Dosyn, D.: Model of Touristic Information Resources Integration According to User Needs, 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2018 – Proceedings 2, 113-116 (2018)
34. Rusyn, B., Lytvyn, V., Vysotska, V., Emmerich, M., Pohreliuk, L.: The Virtual Library System Design and Development, *Advances in Intelligent Systems and Computing*, 871, 328-349 (2019)
35. Lytvyn, V., Vysotska, V., Dosyn, D., Burov, Y.: Method for ontology content and structure optimization, provided by a weighted conceptual graph, *Webology*, 15(2), 66-85 (2018)
36. Lytvyn, V., Peleshchak, I., Vysotska, V., Peleshchak, R.: Satellite spectral information recognition based on the synthesis of modified dynamic neural networks and holographic data processing techniques, 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2018 – Proceedings 1, 330-334 (2018)
37. Gozhyj, A., Kalinina, I., Vysotska, V., Gozhyj, V.: The method of web-resources management under conditions of uncertainty based on fuzzy logic, 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2018 – Proceedings 1, 343-346 (2018)
38. Lytvyn, V., Vysotska, V., Dosyn, D., Lozynska, O., Oborska, O.: Methods of Building Intelligent Decision Support Systems Based on Adaptive Ontology. In: *International Conference on Data Stream Mining and Processing, DSMP 2018*, 145-150 (2018)
39. Rusyn, B., Vysotska, V., Pohreliuk, L.: Model and architecture for virtual library information system, 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2018 – Proceedings 1, 37-41 (2018)
40. Vysotska, V., Lytvyn, V., Burov, Y., Berezin, P., Emmerich, M., Basto Fernandes, V.: Development of Information System for Textual Content Categorizing Based on Ontology. In: *CEUR Workshop Proceedings, Vol-2362*, 53-70 (2019)
41. Burov, Y., Vysotska, V., Kravets, P. Ontological approach to plot analysis and modeling. *CEUR Workshop Proceedings, Vol-2362*, 22-31