КОНЦЕПЦИЯ ПРОЕКТА

*Команда:* RBSoft

*Название проекта:* Project123

*Цель:* автоматизация сбора данных на танцевальных соревнованиях.

**Задачи**

Создание сервера, приложения для ведущего соревнования и судейской коллегии.

**Участники**

Михайлуца Елена – Product owner, Designer

Романенко Юрий – Scrum master, QA

Ничипорчук Артем – Developer

Причиненко Максим – Developer

Ковалев Антон – Developer

**Описание проекта**

Программный проект для улучшения проведения танцевальных соревнований, уменьшает время на обработку информации, уменьшает влияние человеческого фактора на результат оценивания.

**Категория пользователей**

Ведущий соревнований, судейская коллегия, скрутинер.

**Ожидаемая нагрузка**

Площадки - не больше 5.

Судьи на площадке - не больше 21.

Ведущий – 1 на площадке.

**Функционал системы**

Система состоит из следующих компонентов:

1. Сетевой ресурс
	1. Хранение данных для работы системы.
2. Приложения на планшете ведущего
	1. Авторизация и выход из системы с подтверждением;
	2. Вывод программы соревнований (список категорий с турами)
	* категории активны при обработанных результатах судей;
	* не обработанные категории не активны;
	* при выборе активной категории выводится список ее судей;
	* возможность возврата из выбранной категории к общему списку и наоборот.
	1. Контроль изменений на площадках
	* участник добавляется в последний заход категории;
	* удаление участника из списка танцоров категории осуществляется, если он не вышел на площадку. Подтверждение удаления обязательно;
	1. Отправление информации на планшеты судейской коллегии.
3. Приложение на планшете судей
	1. Авторизация и выход из системы с подтверждением;
	2. Принятие информации от ведущего
	* отборочный тур
		+ название категории;
		+ тур (1/32, 1/16, 1/8, 1/4, 1/2);
		+ индекс судьи (A, B, C и т.д.);
		+ фамилия, имя судьи;
		+ список всех участников категории (max количество танцоров в одном заходе - 15).
	* финал
		+ название категории;
		+ тур (финал);
		+ индекс судьи (A, B, C и т.д.);
		+ фамилия, имя судьи;
		+ таблица: 1 столбец – номера участников, 2 – места;
		+ таблица для выбора мест.
	1. Отправление готового результата на сервер.

**Целевые платформы**

Сетевой ресурс - Windows 7

Приложения - Android

**Прототипы пользовательского интерфейса**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|

|  |
| --- |
| Личность 1 |
| Личность 2 |
| Личность 3 |
|  |
|  |
|  |
|  |
|  |

 |

Рис. 1 Выбор пользователя (из списка ведущих и судей)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|

|  |  |
| --- | --- |
| Судья | Фамилия Имя |
| Пароль |  |

|  |  |
| --- | --- |
| Отмена | Войти |

 |

Рис. 2 Авторизация пользователя



Рис. 3 Программа соревнований



В случае соревнований по бальным танцам: в столбце «Танец» виды танцев перечисляются через запятую.

Рис. 4 Бегунок ведущего для отборочных туров

Рис. 5 Бегунок ведущего для финалов

Рис. 6 Бегунок судьи для отборочных туров

Рис. 7 Бегунок судьи для финалов

Примечание:

Бегунок для всех финалов: простых и Золотых. В случае Золотых финалов кол-во изменяется с 1 на указанное кол-во в файле.

**Языки программирования**

Java

**Библиотеки и фреймворки**

Android SDK, Netty.

**Протоколы**

Протоколы передачи данных на основе TCP.
Каждое сообщение между клиентом и сервером будет представляться в такой форме:

*[тип\_сообщения][размер\_тела][тело\_сообщения]*

*[тип\_сообщения]* = 1 байт, указывает на тип сообщения
*[размер\_тела]* = 4 байта, указывает на длину тела сообщения
*[тело\_сообщения]* = *[размер\_тела]* байт, содержит сообщение

При этом *[тело\_сообщения]* - это последовательность байт - представление объекта сообщения после сериализации. В зависимости от типа, этот объект может содержать различные наборы полей. Для сериализации и десериализации объектов, скорее всего, будет использоваться Protocol Buffers.

**Риски проекта**

1. Неизвестная предметная область;
2. Отсутствие ресурсов для тестирования приложений.