

Схожая проблема возникает и при реализации взаимодействия Smart TV телевизора и разрабатываемого приложения с клавишами на пульте телевизора. Каждая операционная система Smart TV предоставляет свой собственный набор кодов кнопок, которые приложение должно правильно обрабатывать: определять, к какой операционной системе они относятся, и реагировать должным образом. Примеры разнообразия кодов клавиш можно увидеть на ресурсах [2] и [3].

Часто при реализации веб-приложений для Smart TV приходится делать проверки на наличие интернет-соединения, определение готовности телевизора к взаимодействию с приложением или готовности осуществить выход из приложения. Поскольку многие телевизоры по-разному это осуществляют, разработчикам приходится писать немало программного кода, учитывающего все особенности каждой поддерживаемой телеплатформы для успешной реализации этих функций.

Решение этих проблем достигается путем разработки и реализации унифицированного интерфейса, который инкапсулирует логику взаимодействия приложения с конкретной версией телевизора, распознаёт текущую платформу Smart TV, и предоставляет разработчикам открытый интерфейс, позволяющий реализовать функции для работы со Smart TV телевизорами без углубления в работу самого интерфейса.

Унифицированный интерфейс реализует сервис, который позволяет создать видео-плеер, управление приложением при помощи контроллеров мыши и клавиатуры, осуществлять различные проверки телевизора на работоспособность, доступ к сети и многое другое. При его использовании, сервис определяет внутри себя текущую платформу и операционную систему, на которой выполняется приложение, и на основании этих определений производит, характерные для данного телевизора инструкции.

Разработанный унифицированный Smart TV интерфейс позволяет абстрагироваться от знаний о конкретных интерфейсах различных платформ и операционных систем Smart TV телевизоров и позволяет полностью сконцентрироваться на решении бизнес-проблем, что является важной его особенностью.

**Список использованных источников:**

1. Википедия [Электронный ресурс]. - Электронные данные. - Режим доступа: [https://ru.wikipedia.org/wiki/Smart\\_TV](https://ru.wikipedia.org/wiki/Smart_TV).
2. Handling Control Key Events | Samsung Developers [Электронный ресурс]. - Электронные данные. - Режим доступа: <http://developer.samsung.com/tv/develop/legacy-platform-library/art00046/index>.
3. LG | webOS TV Developer | Remote Control [Электронный ресурс]. - Электронные данные. - Режим доступа: <http://webostv.developer.lge.com/design/webos-tv-system-ui/remote-control>.

## **ПРОГРАММНОЕ СРЕДСТВО «ОБЩЕРЕСПУБЛИКАНСКИЙ БАНК ДАННЫХ УЧАСТНИКОВ РЕПЕТИЦИОННОГО ТЕСТИРОВАНИЯ» С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ ASP .NET CORE**

*Малец И.В., Тимофеев Н.И., Зайкина И.С.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Парамонов А.И. – к.т.н., доцент*

На первых этапах проведения репетиционного тестирования каждый пункт тестирования реализовывал свои возможности автоматизации данного бизнес-процесса. Это усложняло возможность централизованного управления репетиционным тестированием. Перечень функциональных возможностей каждой системы так же отличался. Поэтому возникла потребность в создании единого программного средства, которое могли бы использовать все пункты тестирования для организации и проведения репетиционного тестирования.

Эпоха новых информационных и коммуникационных технологий принесла значительные перемены в сферу производства, образования и деловой активности людей. Сферу образования можно назвать флангом, который принимает на себя основной удар. Одним из таких направлений является организация проведения репетиционного тестирования (РТ), как составляющей всего этапа проведения проверки знаний будущих абитуриентов. Обязанности организации и проведения РТ возложены на «Республиканский институт контроля знаний».

Предлагается решение в виде информационной системы, которая позволит организовать работу в рамках «одного окна», и осуществлять управление пунктами регистрации и проведения РТ.

В ходе проведения анализа предметной области, были выделены основные функциональные требования, предъявляемые к программному средству (ПС):

- филиальная возможность работы;
- регистрация новых пунктов;

- организация доступа к регистрации;
- формирования списков зарегистрированных пользователей;
- контроль явки на РТ;
- личный кабинет участника РТ;
- отслеживание статуса регистрации;
- формирование и подтверждение актов приема-передачи бланков ответов;
- осуществление платежей в режиме реального времени с помощью системы «Расчет»;
- оформления пропусков;
- экспорт и импорт данных из Excel-файла;
- формирование сводной ведомости;
- генерация статистики по репетиционному тестированию;
- загрузка результатов репетиционного тестирования.

На основании списка требований построена функциональная модель программного средства в нотациях IDEF0 [1], а также диаграмма вариантов использования (рисунок 1).

Были выделены следующие действующие лица: главный администратор, администратор, пользователь. Для разрабатываемой системы, исходя из потребностей действующих лиц, выделены следующие варианты использования: вход в систему, просмотр списка пунктов тестирования (ПТ), изменение ПТ, добавление ПТ, загрузка результатов, выбор файла результатов, подтверждение актов приёма-передачи бланков ответов, просмотр сеансов тестирований, изменение сеанса тестирования, просмотр списка зарегистрированных, печать ведомости, отметка явившихся, добавление сеанса тестирования, регистрация в системе, ввод своих данных, просмотр списка регистраций, отмена регистрации, просмотр подробной информации, регистрация на РТ.

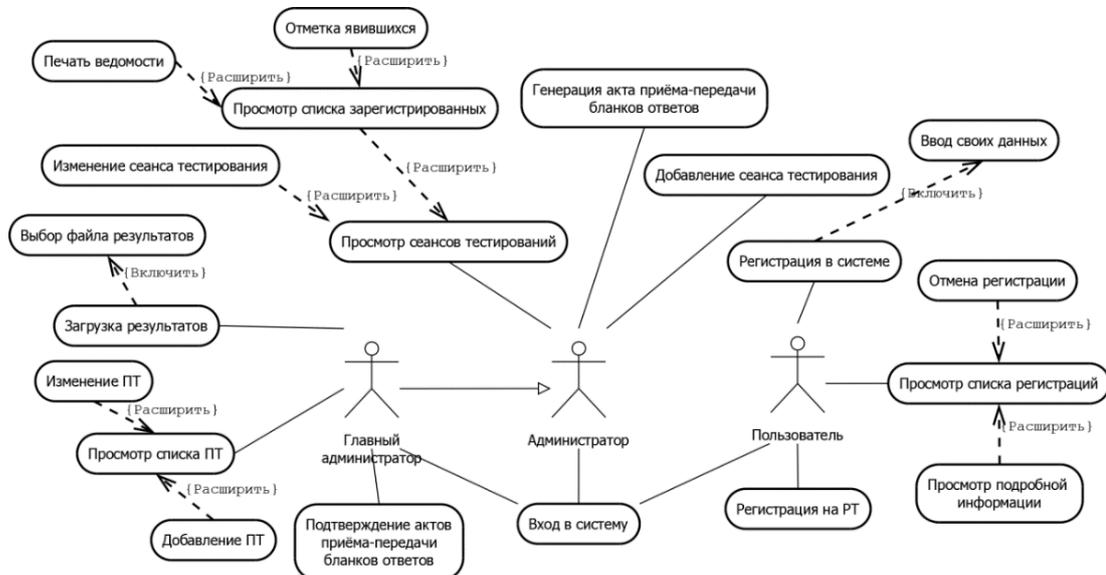


Рисунок 1 – Диаграмма вариантов использования

В качестве языка программирования для серверной части приложения был выбран C#, а именно фреймворк ASP.NET Core [2]. В качестве языка программирования для клиентской части приложения был выбран Typescript, фреймворк Angular [3]. Для хранения информации будет использовано СУБД Microsoft SQL Server 2017. Для разработки программного средства «Общереспубликанский банк данных участников репетиционного тестирования» будет использоваться среда разработки Visual Studio 2017.

Интерфейс программного средства представлен на рисунке 2.

Реализована возможность многопользовательского режима работы. Разработан понятный и простой интерфейс для работы с программным средством с учетом особенностей предметной области.

Выходными данными программного средства являются: обобщенная ведомость участников репетиционного тестирования, отображаемая в виде файла pdf; файл Microsoft Excel обобщенных данных участников тестирования, зарегистрированных на конкретную дату; пропуск для прохождения тестирования в выбранном пункте в виде файла pdf. Обобщенные сведения доступны только администраторам пунктов тестирования. Печать пропуска доступна всем участникам тестирования, которые произвели оплату созданной регистрации на выбранный предмет и выделенную дату.

 МС 2151450 Выйти



Тимофеев Никита Игоревич

Лицевой счёт для оплаты: **0000002**

Количество свободных оплат: **1**

✎ Редактировать

Номер регистрации	Дата	Название ПТ	Место	Предмет	Статус	
0000013	12.12.2018 14:00	ГУО «Средняя школа №4 г. Червеня»	ул. Барыкина, 91а, ФОК	Русский язык Язык теста: русский	Результат 65	<span style="background-color: #007bff; color: white; padding: 2px 5px;">👁 Подробнее</span>
0000049	21.12.2018 14:00	ГУО «Средняя школа №4 г. Червеня»	ул. Барыкина, 91а, ком. 15	Физика Язык теста: русский	Не явился	<span style="background-color: #007bff; color: white; padding: 2px 5px;">👁 Подробнее</span>
0000059	31.12.2018 14:00	УО «Республиканский институт контроля знаний»	Оранжевое здание, 2-ой этаж, к. 27	Русский язык Язык теста: русский	Оплачено	<span style="background-color: red; color: white; padding: 2px 5px;">✖ Отменить</span> <span style="background-color: #007bff; color: white; padding: 2px 5px;">👁 Подробнее</span> <span style="background-color: #007bff; color: white; padding: 2px 5px;">🚪 Пропуск</span>
0000060	29.12.2018 14:00	ГУО «Средняя школа №4 г. Червеня»	ул. Барыкина, 91а, акт. зал	Физика Язык теста: белорусский	Ожидается оплата	<span style="background-color: red; color: white; padding: 2px 5px;">✖ Отменить</span> <span style="background-color: #007bff; color: white; padding: 2px 5px;">💳 Оплатить</span> <span style="background-color: #007bff; color: white; padding: 2px 5px;">👁 Подробнее</span>

Рисунок 2 – Домашняя страница пользователя

Внедрение разработанного программного средства автоматизировало процесс регистрации на РТ во всех пунктах тестирования, а также облегчило работу организаторов репетиционного тестирования в пунктах его проведения. Личный кабинет участника РТ позволил легче производить регистрацию на репетиционное тестирование, а также отслеживать статус своих регистраций.

Отпала необходимость в проведении различного рода совещаний, командировок, письменных коммуникаций.

**Список использованных источников:**

1. Маклаков, С. В. Инструментальные средства создания корпоративных информационных систем / С. В. Маклаков. – Компьютер Пресс. 1998. – №7.
2. Чамберс Д., Пэккетт Д., Тиммс С., ASP.NET Core. Разработка приложений/ Д. Чамберс – 2018. – 464 с.
3. Файн Я., Моисеев А., Angular и TypeScript. Сайтостроение для профессионалов/ Я. Файн 2018 – 368 с.

## ИНФОРМАЦИОННАЯ СИСТЕМА СБОРА И ОБРАБОТКИ ТЕЛЕМЕТРИЧЕСКИХ ДАННЫХ

*Малец И.В., Тимофеев Н.И., Моженкова Е.В.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Пармонов А.И. – к.т.н., доцент*

В современном мире, наполненном большим количеством информационных систем и технологий, важное место занимает их всесторонний мониторинг. Суть его заключается в сборе необходимых данных и их анализе. Анализ собранной информации об устройстве или программном средстве позволяет предотвратить ошибки или, на основании полученной информации, оптимизировать их производительность. Практически все современные устройства и программные средства имеют собственные системы мониторинга. Но в случае, когда количество систем для мониторинга слишком велико, или к ним нет непосредственного доступа, – стандартных средств становится недостаточно.

Сбор, хранение и обработка телеметрических данных становятся все более актуальными задачами. На сегодня известно уже большое количество систем и программных средств для её решения. Часть из них нацелена на мониторинг конкретных типов систем – программные средства для мониторинга сетей: Observium [1], SolarWinds Network Performance Monitor [2]. Другие, более гибкие, могут работать с большим количеством устройств и протоколов, по средствам добавления в них модулей или установки специальных приложений-агентов. Примером таких систем являются Zabbix [3] или CA UIM [4]. Практически все современные системы сбора и обработки телеметрических данных имеют веб-интерфейс для доступа к данным из любой точки мира. В последнее время многие системы для сбора и обработки телеметрических данных становятся доступны в формате облачных