

Санкт-Петербургский государственный университет

Кафедра Системного Программирования

Группа 21.Б10-мм

# Реализация механизма приглашения экспертов в сервис HwProj 2.0.1

***КСЕНЧИК Никита Владимирович***

Отчёт по учебной практике  
в форме «Решение»

Научный руководитель:  
старший преподаватель кафедры СП, к.т.н., Литвинов Ю. В.

Консультант:  
инженер-программист, Бережных А. В.

Санкт-Петербург  
2024

# **Оглавление**

|                             |           |
|-----------------------------|-----------|
| <b>Введение</b>             | <b>3</b>  |
| <b>1. Постановка задачи</b> | <b>4</b>  |
| <b>2. Обзор</b>             | <b>5</b>  |
| <b>3. Сбор требований</b>   | <b>10</b> |
| <b>4. Метод</b>             | <b>11</b> |
| <b>5. Реализация</b>        | <b>13</b> |
| <b>6. Апробация</b>         | <b>20</b> |
| <b>Заключение</b>           | <b>22</b> |
| <b>Список литературы</b>    | <b>23</b> |

# Введение

В современном образовательном процессе применяются разнообразные системы управления обучением. В СПбГУ, например, активно используются Microsoft Teams и Blackboard Learn. Эти платформы, несмотря на свою популярность, имеют некоторые недостатки. В результате студенты матмеха создали HwProj 2.0.1, сервис, который позволяет сдавать домашние задания по программированию и используется в некоторых преподаваемых курсах уже не один год.

В курсах обычно участвует множество студентов, а объем домашних заданий значителен. Проверка такого количества работ становится трудной задачей для одного преподавателя, поэтому приглашаются сторонние помощники. Эти помощники могут быть старшекурсниками или специалистами из других компаний. Эксперты не только помогают преподавателям, но и делятся своим опытом со студентами, проверяя их задания.

Человеку, приглашенному для проверки нескольких задач, важно иметь возможность быстро перейти к курсу и сразу сосредоточиться на домашних заданиях, не углубляясь в особенности курса и не сталкиваясь с ненужными элементами интерфейса HwProj.

В настоящее время для этих задач в сервисе используется роль преподавателя. Она позволяет пользователю заходить на курс и проверять домашние задания. Однако интерфейс преподавателя содержит множество ненужных элементов, таких как список заявок на вступление, перечень всех курсов, доступ ко всем домашним заданиям и всем студентам, хотя эксперту нужна только часть этой информации. Чтобы облегчить работу экспертам, было решено разработать новую функциональность для их приглашения, которая будет более удобной.

Для правильного отображения курса эксперта необходимо добавить механику прокси-курсов: исходя из того, какой срез курса нужен эксперту, фильтровать домашние работы/студентов и возвращать урезанный «прокси-курс».

# 1. Постановка задачи

Требуется реализовать механизм фильтрации курсов и подготовить рабочее пространство эксперта для взаимодействия с сервисом. Для достижения цели были поставлены следующие задачи:

1. Опросить преподавателей и собрать требования к реализуемой функциональности.
2. Внедрить механизм работы с «прокси-курсами».
3. Подготовить рабочее пространство эксперта для взаимодействия с сервисом.
4. Провести апробацию по методике SUS (System Usability Scale) [1] на преподавателях СПбГУ с целью измерения степени удобства использования реализованной функциональности со стороны преподавателя и эксперта.

## 2. Обзор

Следующие технологии уже были использованы при разработке сервиса HwProj 2.0.1 и остались без изменений:

- веб-фреймворк ASP.NET Core с паттерном Model-View-Controller [3];
- язык программирования C# для серверной части HwProj 2.0.1;
- TypeScript и библиотека React для создания фронтенда [4];
- база данных Microsoft SQL Server.

### Приглашение эксперта

Работа над функциональностью данного раздела была выполнена сокомандником.

В меню эксперта был добавлен новый элемент «К списку экспертов» (рис. 1). При нажатии на него пользователь попадает на страницу «Книги экспертов».

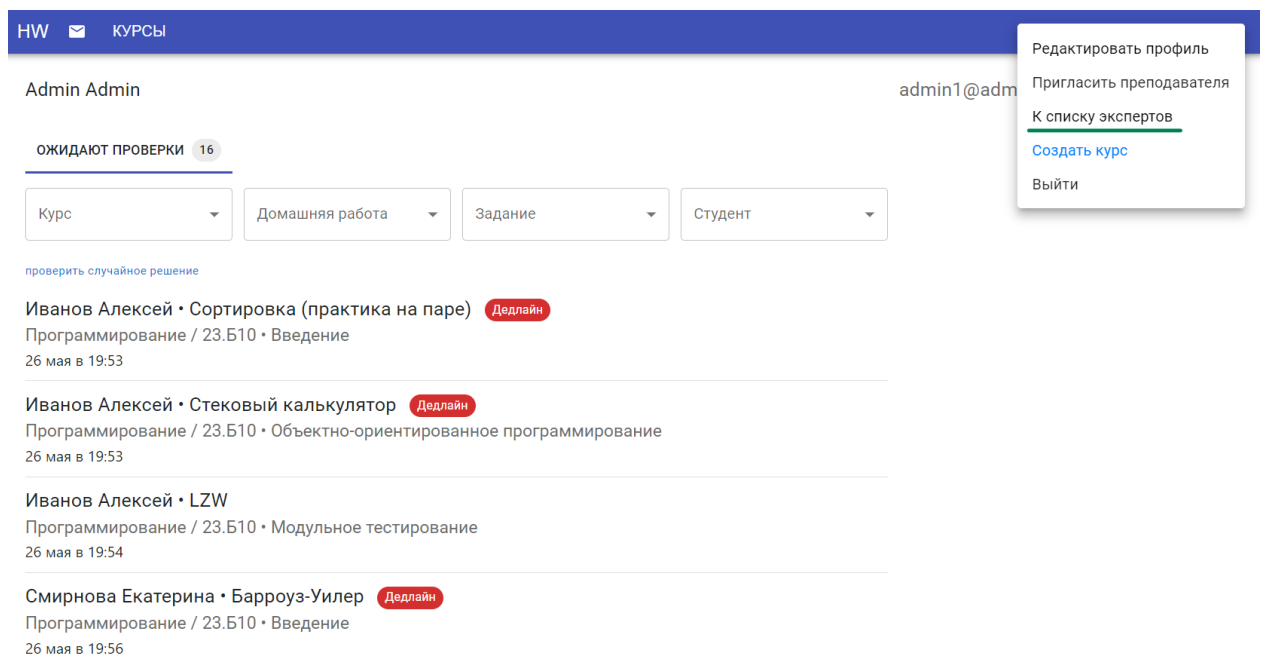


Рис. 1: Меню преподавателя

| <div> <div>HW</div> <div>✉</div> <div>КУРСЫ</div> <div>☰</div> </div> |                          |              |                                                                                                                |
|-----------------------------------------------------------------------|--------------------------|--------------|----------------------------------------------------------------------------------------------------------------|
| Эксперты                                                              |                          |              | <input checked="" type="checkbox"/> Показать всех                                                              |
| ФИО                                                                   | Почта                    | Компания     | Тэги                                                                                                           |
| Чижиков Дмитрий Владимирович                                          | chizikov.dmitriy@mail.ru | Ланит-Терком |                                                                                                                |
| Иванов Иван Иванович*                                                 | ivan.ivanov@mail.ru      | JetBrains    | <div>Java</div> <div>Kotlin</div> <div>Maven</div>                                                             |
| Дебиан Центос Федорович                                               | admin@linux.com          | Canonical    | <div>C++</div> <div>Linux</div>                                                                                |
| Кириленко Яков Александрович*                                         | i.kirilenko@spbu.ru      | YADRO        | <div>C++</div> <div>Kotlin</div> <div>QT</div> <div>Android</div> <div>C</div> <div>Linux</div> <div>git</div> |
| <div>+</div>                                                          |                          |              |                                                                                                                |

Рис. 2: Книга экспертов

В книге отображаются эксперты, добавленные пользователем (рис. 2). При нажатии на кнопку «Показать всех» можно увидеть список всех экспертов. Эксперты, добавленные пользователем, отмечены синей звездочкой, при наведении на которую появляется подсказка.

При нажатии на плюс в левом нижнем углу появляется окно регистрации нового эксперта (рис. 3). В данном окне необходимо ввести ФИО эксперта и его почту. Так же можно добавить дополнительную информацию:

- компания – место, в котором работает эксперт;
- тэги – пометки, по которым можно понять, в какой области эксперт специализируется;
- био – дополнительная информация об эксперте, которая, на данный момент, нигде не отображается, но в будущем будет доступна для студентов и преподавателей.

При нажатии на кнопку «Зарегистрировать» появляется надпись «Эксперт успешно зарегистрирован» (рис. 4). С этого момента преподаватели смогут найти эксперта в «Книге экспертов».

HW ✉ КУРСЫ

Эксперты

Показать всех

ФИО

Чижиков Дмитрий Владимирович chizikov.dmitriy

Иванов Иван Иванович ivan.ivanov

Дебиан Центос Федорович admin@li

Кириленко Яков Александрович i.kirilenko

Почта

Python C# ООП Добавить тэг

ПРИГЛАСИТЬ

Зарегистрировать эксперта

Имя \* Иван

Фамилия \* Сидоренко

Отчество Романович

Электронная почта \* ivan.sidorenko@gmail.com

Компания Yandex

Тэги Python C# ООП Добавить тэг

Дополнительная информация (био) Выпускник матмеха СПбГУ 2015

ЗАКРЫТЬ ЗАРЕГИСТРИРОВАТЬ

Рис. 3: Регистрация эксперта

HW ✉ КУРСЫ

Эксперты

Показать всех

ФИО

Чижиков Дмитрий Владимирович chizikov.dmitriy

Иванов Иван Иванович ivan.ivanov

Дебиан Центос Федорович admin@li

Кириленко Яков Александрович i.kirilenko

Почта

Python C# ООП Добавить тэг

ПРИГЛАСИТЬ

Зарегистрировать эксперта

Эксперт успешно зарегистрирован

Имя \* Иван

Фамилия \* Сидоренко

Отчество Романович

Электронная почта \* ivan.sidorenko@gmail.com

Компания Yandex

Тэги Python C# ООП Добавить тэг

Дополнительная информация (био) Выпускник матмеха СПбГУ 2015

ЗАКРЫТЬ ЗАРЕГИСТРИРОВАТЬ

Рис. 4: Эксперт зарегистрирован

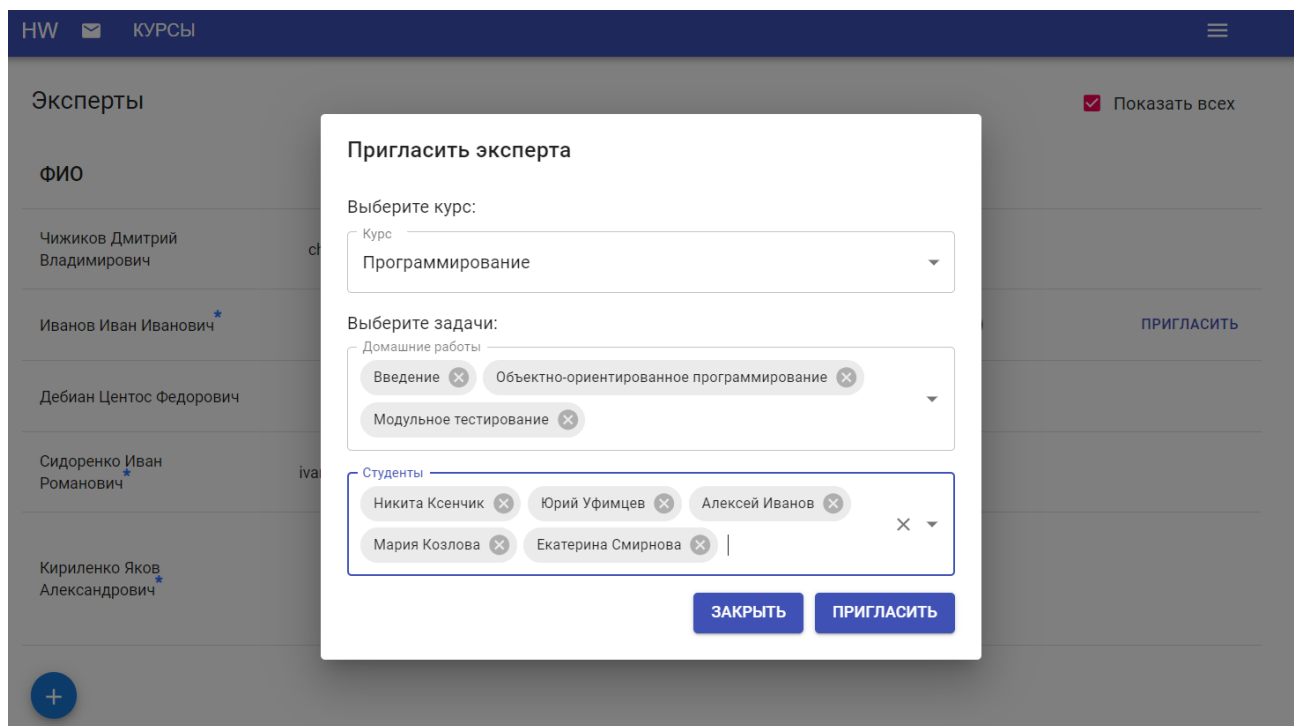


Рис. 5: Приглашение эксперта на курс

При нажатии кнопки «Пригласить» в строке с экспертом появляется окно приглашения эксперта на курс (рис. 5). Здесь нужно указать курс, домашние работы и студентов. Если оставить поля «Домашние работы» и «Студенты» пустыми, то эксперту будут доступны все имеющиеся задания и студенты курса.

При нажатии на кнопку «Пригласить» лектору выдается ссылка с JWT токеном (рис. 6) [5]. Время действия токена указано под ссылкой. Данную ссылку нужно передать эксперту. При приглашении эксперта на другой курс преподаватель получит эту же ссылку. Эксперту автоматически добавится новый курс.

При первом переходе по ссылке эксперт попадает на страницу, где может отредактировать данные, введенные преподавателем (рис. 7). После нажатия кнопки «Редактировать профиль» эксперт попадет в своё рабочее пространство и сможет начать работу.

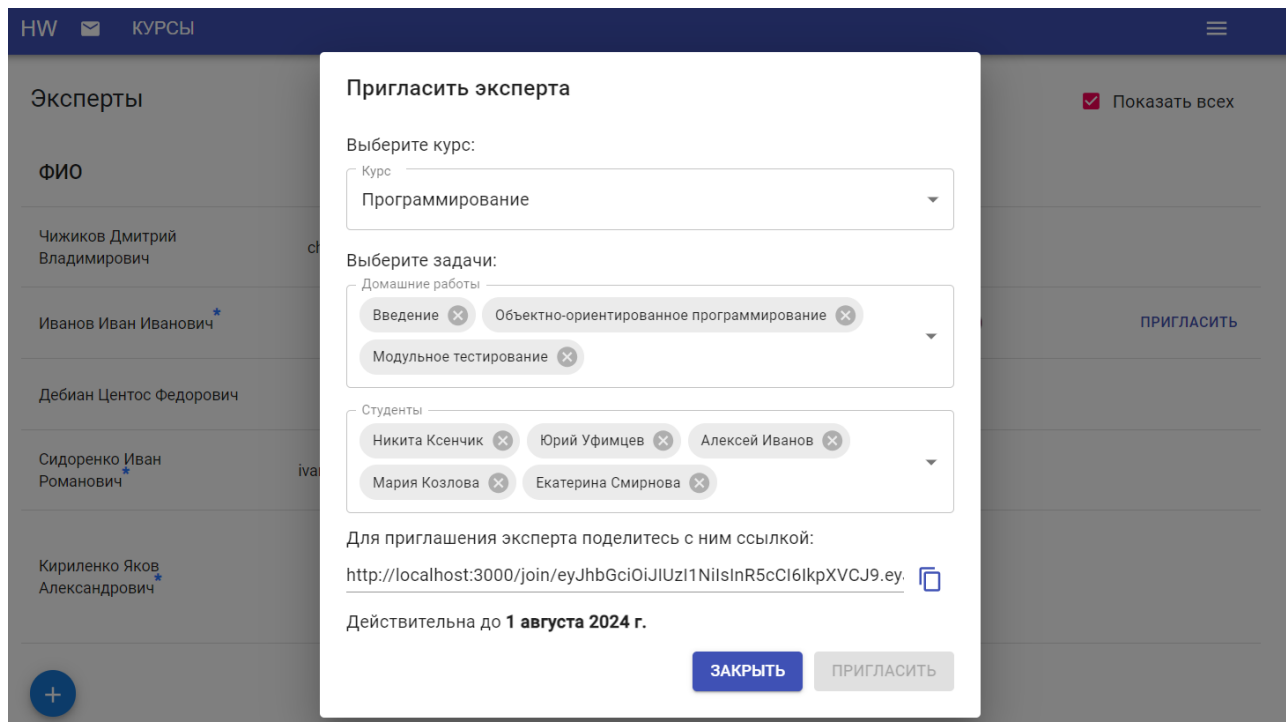


Рис. 6: Получение ссылки для эксперта

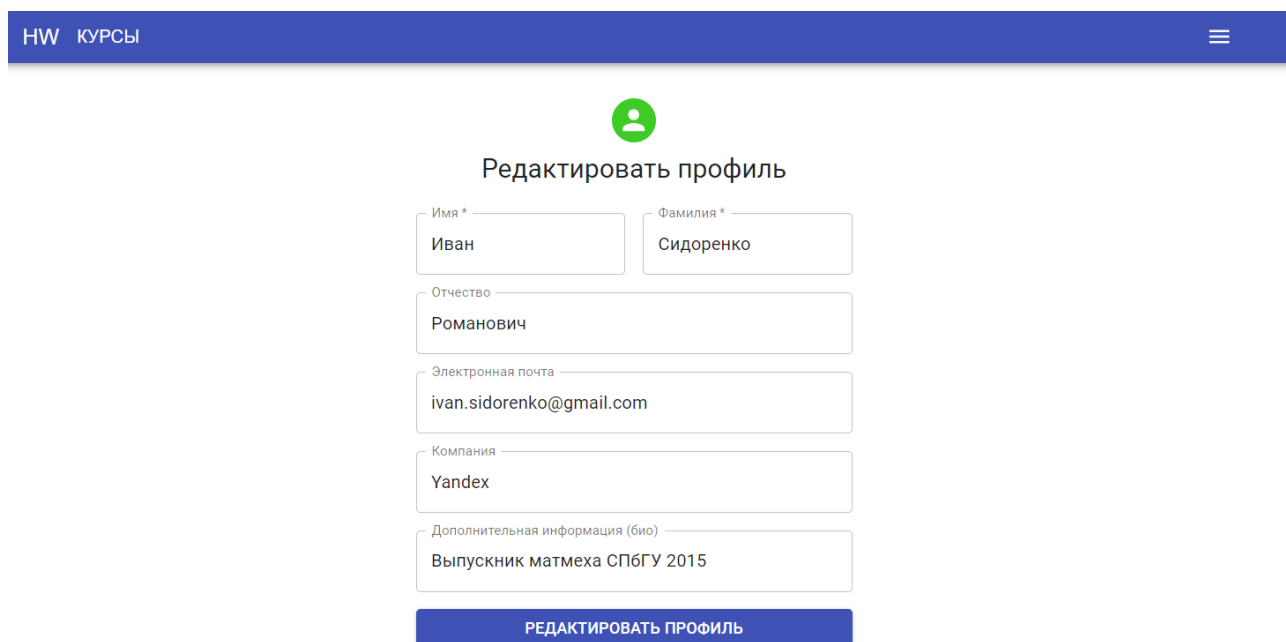


Рис. 7: Стартовая страница эксперта

### 3. Сбор требований

Для более четкого понимания требований к новой функциональности было решено провести опрос преподавателей, имеющих опыт работы с HwProj.

В опросе, посвященном выявлению требований к функциональности по приглашению экспертов, приняли участие четыре преподавателя.

Проанализировав полученную информацию, были выделены следующие требования:

1. В рабочем пространстве эксперта не должно быть лишней информации:
  - информация о подающих заявку на курс;
  - вкладка «Все курсы»;
  - вкладка «Уведомления»;
  - разделы меню «пригласить преподавателя», «создать курс», «пригласить эксперта».
2. Эксперту должны быть доступны только назначенные ему домашние задания и студенты.
3. Во вкладке «Мои курсы» при переходе на курс эксперт должен сразу попадать к списку непроверенных решений.

## 4. Метод

Проект выполнялся в команде из двух человек: автором данной работы и Уфицевым Юрием Дмитриевичем.

После сбора требований было решено распределить задачи следующим образом. Юрий получил следующие задачи:

- создание «Книги экспертов»;
- добавление роли «Эксперт»;
- реализация функциональности регистрации аккаунта эксперта в системе;
- создание профиля эксперта с возможностью его редактирования.

Автору данной работы достались задачи, связанные с работой эксперта в «Пространстве эксперта», а именно удаление лишних элементов интерфейса и добавление логики фильтрации курсов для экспертов.

### Прокси-курсы

Роль эксперта сужает роль преподавателя, ему также доступна проверка решений сданных студентами. Однако эксперту доступна лишь часть проверяемого курса: часть студентов и часть домашних работ. Для того чтобы хранить и использовать эти данные, было принято решение создать механизм фильтрации курсов и создания «прокси-курсов» на основе данных фильтров.

Логика работы «прокси-курсов»:

1. при добавлении эксперта на курс создаётся фильтр;
2. при запросе курса экспертом применяется ранее созданный фильтр и возвращается «прокси-курс» с урезанной областью видимости.

Данный механизм позволяет легко менять содержание курса для разных пользователей, при этом требует внести небольшое количество

изменений в код. В дальнейшем его можно будет использовать для похожих задач:

- разделение студентов на курсе между преподавателями и помощниками преподавателя (Teacher Assistant);
- выделение задач на конкретных студентов.

## 5. Реализация

### Фильтры курсов

Для фильтрации курса необходимо хранить идентификатор студентов, домашних заданий и менторов курса. Это можно было сделать несколькими способами:

1. Создать сущности Student, Homework, Mentor, и в них хранить идентификатор эксперта и идентификатор курса, которому они принадлежат. Данный метод требует создания новых таблиц, приводит к возникновению множества дублирующей информации и неоправданно усложняет структуру базы данных.
2. Сделать класс для фильтров курсов, поля под каждый список и хранить эти поля в базе, например записывая элементы через '/'. Данный способ будет неустойчив к изменениям: если потребуется менять структуру фильтров, добавлять одни поля, убирать другие, то возникнут проблемы с миграциями и обратной совместимостью, будут возникать неиспользуемые поля из предыдущих версий, которые нужно будет поддерживать.

В итоге было принято решение реализовать фильтры следующим образом:

- Класс Filter – в нем содержится вся необходимая информация для фильтрации курса.
- Класс CourseFilter содержит три поля:
  - Id – уникальный идентификатор для хранения записи в базе.
  - Filter – сам фильтр, который используется в работе сервиса. Данное поле не хранится в базе данных.
  - FilterJson – поле, которое помогает осуществлять связь между фильтром в базе данных и на бэкенде. При записи в базу оно

берет данные из поля Filter, сериализует и сохраняет как строку. При получении сущности CourseFilter из базы FilterJson десериализует строку и записывает ее в поле Filter. Более поздние версии EntityFramework позволяют производить сериализацию/десериализацию полей автоматически, однако в имеющейся версии пришлось выделять под такую работу дополнительное поле [2].

На рисунке 8 приведены базовые классы связанные с фильтрами.

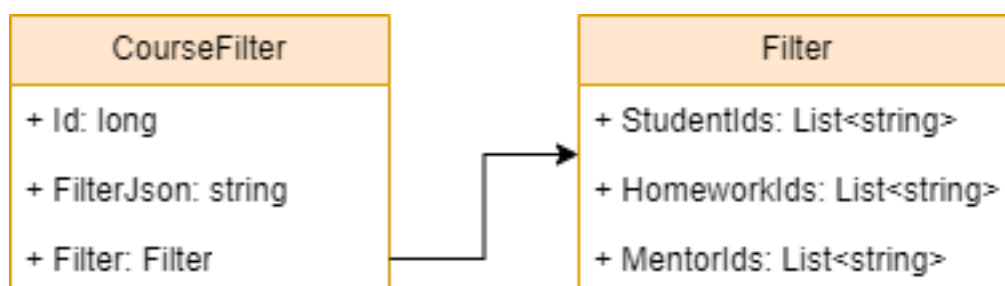


Рис. 8: Фильтры курсов

Данная схема хранения фильтров обладает следующими преимуществами:

- Удобная работа с фильтром на бэкенде: мы сразу имеем Filter со списками, который уже можно применять при осуществлении логики фильтрации.
- Устойчивость к изменениям фильтров: при добавлении новых полей старые записи останутся актуальными и не потребуют внесения изменений.

Данная схема имеет свои недостатки: при необходимости массового изменения данных в записях придется писать миграторы, которые произведут данные действия.

Плюсы третьего метода на данном этапе развития HwProj 2.0.1 показали значительно больше минусов, поэтому он и был выбран при реализации.

Сущности фильтров в базе данных:

Сущность «UserToCourseFilter» представляет собой абстракцию связи между пользователем на курсе и фильтром.

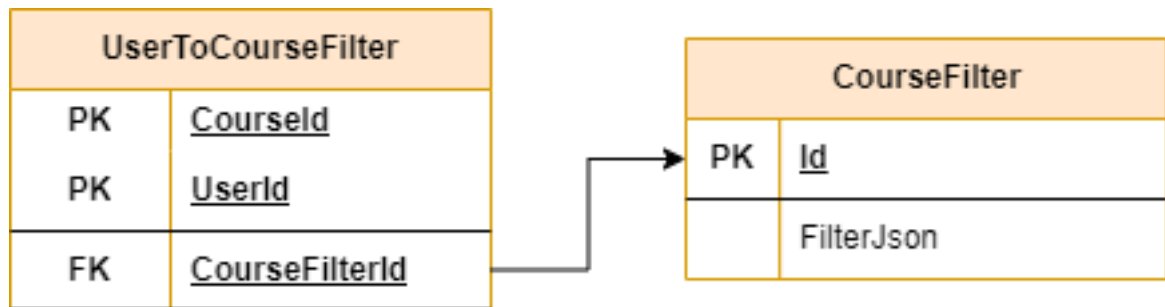


Рис. 9: Сущности фильтров курсов

## Работа с фильтрами

На рисунке 10 представлены сервисы и репозитории для работы с фильтрами:

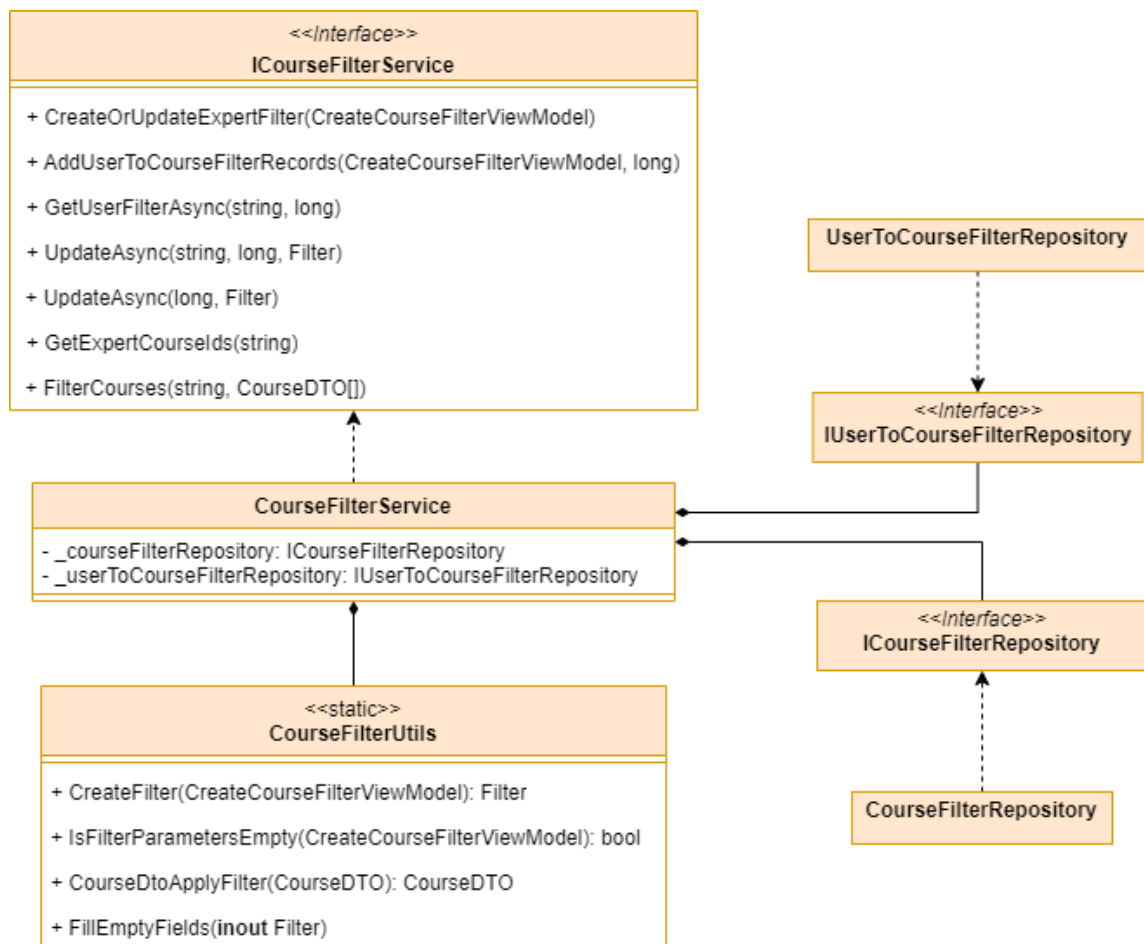


Рис. 10: Сервисы и репозитории для работы с фильтрами

Рассмотрим их подробнее:

- **UserToCourseFilterRepository** содержит методы CRUD для класса **UserToCourseFilter**.

- CourseFilterRepository содержит методы CRUD для класса CourseFilter.
- CourseFilterUtils содержит методы для применения фильтров: создание фильтра по полученным данным, применение фильтра к курсу, заполнение пустых полей фильтра, проверка параметров создания фильтра на валидность.
- CourseFilterService – класс, через который другие сервисы могут работать с фильтрами: создавать фильтры, получать фильтры, получать информацию по фильтрам.

Создание фильтра происходит следующим образом:

1. На фронте заполняются параметры создаваемого фильтра.
2. Параметры через цепочку сервисов и контроллеров доходят до CourseFilterService.
3. Через CourseFilterUtils параметры проверяются и создаётся фильтр.
4. Данный фильтр и связи UserToCourseFilter сохраняются в базу.

Фильтрация курса происходит следующим образом:

1. До CourseService доходит запрос на получение курса.
2. CourseService собирает все нужные данные для данного курса.
3. Через CourseFilterService запрашивается фильтр для пользователя на данный курс.
4. На курс накладывается фильтр.
5. Пользователю возвращается отфильтрованный курс.

Данная схема работы не завязана на фильтрах именно для экспертов. Если в будущем добавятся фильтры для других ролей, то схема работы сервиса останется прежней.

## Рабочее пространство эксперта

Работа над функциональностью данного раздела была выполнена автором данной работы.

После страницы с редактированием данных или же при втором и более переходе по ссылке эксперт попадает на начальную страницу HwProj (рис. 11). Здесь он может увидеть список непроверенных решений.

Получение данных данной страницы для преподавателя происходит следующим образом:

1. Запрашиваются все курсы преподавателя.
2. С курсов достаются все домашние работы.
3. Из базы данных берутся все непроверенные решения всех студентов по данным домашним работам и возвращаются на фронтенд.

Так как преподавателям всегда доступны все домашние работы и студенты, для них такая схема приемлема. Для эксперта могут быть доступны несколько курсов, студенты на курсах могут пересекаться, поэтому данная схема была модифицирована.

1. Запрашиваются все курсы эксперта (курсы приходят уже отфильтрованными).
2. Группируются домашние работы и студенты каждого курса.
3. По каждому курсу происходит асинхронный запрос непроверенных решений в SolutionService.
4. При помощи `Task.WhenAll()` мы ожидаем возвращения всех решений [6].
5. Полученные решения группируются, сортируются и возвращаются на фронтенд.

Командой было принято решение на данный момент не отсылать никаких уведомлений экспертам, чтобы не захламлять их почту. Вследствие этого с верхней панели была удалена вкладка «Уведомления».

Иван Романович Сидоренко

ivan.sidorenko@gmail.com

ОЖИДАЮТ ПРОВЕРКИ 8

|        |                   |           |           |
|--------|-------------------|-----------|-----------|
| Курс ▾ | Домашняя работа ▾ | Задание ▾ | Студент ▾ |
|--------|-------------------|-----------|-----------|

[проверить случайное решение](#)

Иванов Алексей • Сортировка (практика на паре) **Дедлайн**  
 Программирование / 23.Б10 • Введение  
 26 мая в 19:53

Иванов Алексей • Стековый калькулятор **Дедлайн**  
 Программирование / 23.Б10 • Объектно-ориентированное программирование  
 26 мая в 19:53

Иванов Алексей • LZW  
 Программирование / 23.Б10 • Модульное тестирование  
 26 мая в 19:54

Смирнова Екатерина • Барроуз-Уилер **Дедлайн**  
 Программирование / 23.Б10 • Введение  
 26 мая в 19:56

Рис. 11: Рабочее пространство эксперта

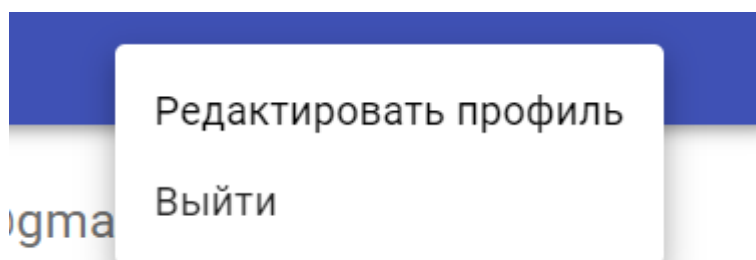


Рис. 12: Меню эксперта

В меню эксперта были удалены все элементы, кроме «Редактировать профиль» и «Выйти» (рис. 12).

Во вкладке «Курсы» (рис. 13) была удалена вкладка «Все курсы» и оставлены только «Мои курсы» и «Завершенные курсы» (отображается только при наличии завершенных курсов).

При нажатии на курс эксперт сразу попадает на страницу непроверенных решений (рис. 14).

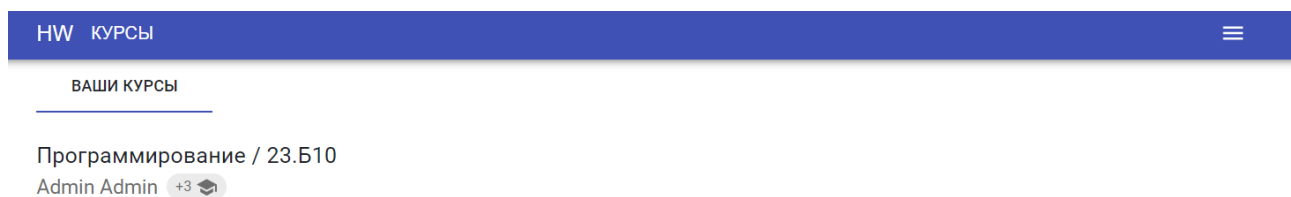


Рис. 13: Курсы

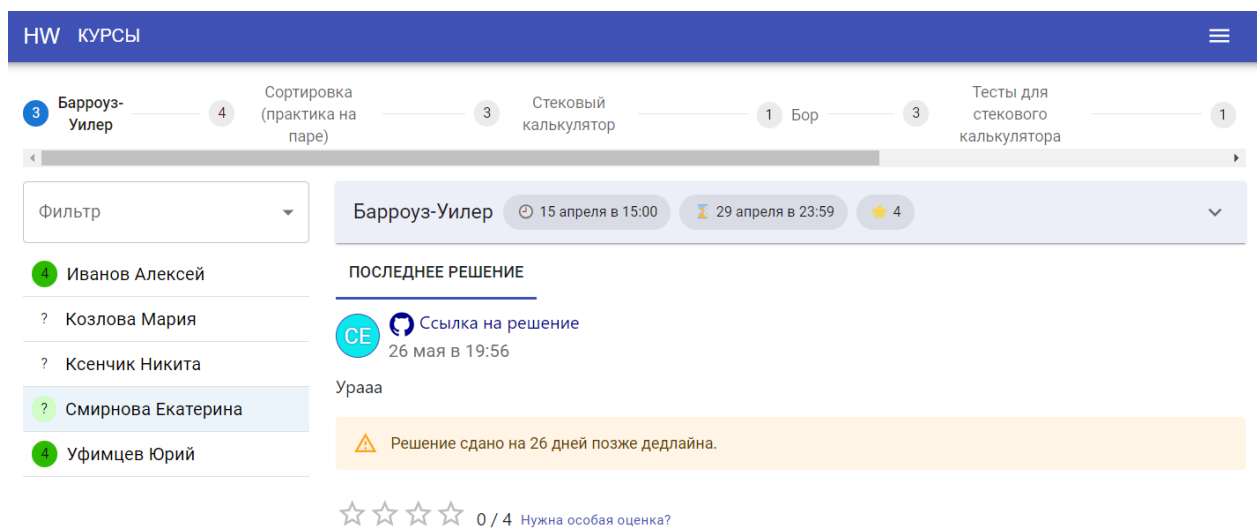


Рис. 14: Страница решений

## 6. Апробация

Презентация выполненной функциональности и сбор отзывов от пользователей проходили поэтапно: подготовка, апробация по методике SUS и анализ полученных данных.

### Подготовка

Были разработаны два тестовых сценария для сбора отзывов:

- Регистрация преподавателем и приглашение эксперта.
- Взаимодействие приглашенного эксперта с сервисом.

На основе выполнения этих действий пользователи заполняли анкету SUS для оценки реализованной функциональности.

### Проведение апробации

Апробацию проводили преподаватели кафедры системного программирования математико-механического факультета СПбГУ: Яков Александрович Кириленко, Юрий Викторович Литвинов.

В ходе апробации были получены следующие предложения по расширению имеющейся функциональности:

- Добавить функциональность приглашения экспертов на странице курса.
- Добавить возможность эксперту не ставить оценку за задачу, а только оставлять свой комментарий, по которому преподаватель курса уже сможет оценить решение.

### Результаты

По методике SUS получен средний балл 91.75. Данный результат указывает на то, что пользователи находят систему интуитивно понятной, эффективной и удобной в использовании.

Существуют следующие угрозы валидности эксперимента:

- маленькая выборка участников;
- знакомство участников с системой.

Несмотря на угрозы валидности, отзывы о существующей функциональности исключительно положительные. Поэтому на данном этапе разработки можно сделать вывод, что проделанная работа соответствует ожиданиям выбранной группы пользователей. Далее данную функциональность следует зафиксировать в главной ветке HwProj и тестировать на всех пользователях сервиса.

## Заключение

В рамках данной работы были внедрены фильтры для курсов и подготовлено рабочее пространство для взаимодействия экспертов с сервисом. Для достижения цели были поставлены и успешно выполнены следующие задачи:

1. Проведён опрос преподавателей для сбора требований к реализуемой функциональности.
2. Внедрён механизм работы с «прокси-курсами».
3. Подготовлено рабочее пространство эксперта для взаимодействия с сервисом.
4. Проведена апробация по методике SUS (System Usability Scale) среди преподавателей СПбГУ, чтобы измерить степень удобства использования реализованной функциональности со стороны преподавателя и эксперта.

Ссылка на пул-реквесты с реализацией:

- <https://github.com/IntelliGeNET/HwProj-2.0.1/pull/379>
- <https://github.com/IntelliGeNET/HwProj-2.0.1/pull/406>
- <https://github.com/IntelliGeNET/HwProj-2.0.1/pull/404>

## Список литературы

- [1] Bhat Adi. SUS. — URL: <https://hr-portal.ru/story/что-такое-shkala-yuzabiliti-sistemy> (дата обращения: 12 июня 2024 г.).
- [2] How to write .NET objects as JSON (serialize). — URL: <https://learn.microsoft.com/en-us/dotnet/standard/serialization/system-text-json/how-to> (дата обращения: 12 июня 2024 г.).
- [3] Model-View-Controller. — URL: <https://www.geeksforgeeks.org/mvc-framework-introduction/> (дата обращения: 12 июня 2024 г.).
- [4] React. — URL: <https://react.dev/> (дата обращения: 12 июня 2024 г.).
- [5] borisyuzhakov. Json Web Tokens (JWT). — URL: <https://habr.com/ru/articles/340146/> (дата обращения: 12 июня 2024 г.).
- [6] Разобраться раз и навсегда: Task.WhenAll или Parallel.ForEachAsync в C#. — URL: <https://habr.com/ru/companies/ruvds/articles/790836/> (дата обращения: 12 июня 2024 г.).