

МИНОБРНАУКИ РФ
ФГБОУ ВО «ИжГТУ имени М.Т. Калашникова»
Кафедра «Программное обеспечение»

Отчет
по лабораторной работе №2
по дисциплине «Проектирование и конструирование
программного обеспечения»
по теме «Разработка базового расписания»

Выполнили
студенты группы Б20-191-2:

Власов Ф.А.
Широбоков А.А.

Принял:

Еланцев М.О.

Ижевск 2023

Сервис по развитию профессиональных навыков сотрудников IT-компаний.

1. Всего на данный момент сервис должен содержать в себе 11 экранных форм:

- Форма регистрации
 - Нужна для регистрации пользователя в системе, содержит в себе поля ввода электронной почты и пароля – эти данные нужны для последующей возможности входа в созданный аккаунт. Пароль нужен также для обеспечения безопасности аккаунта. При отсутствии входа в какой-либо аккаунт загружается по умолчанию при запуске приложения. После завершения регистрации происходит переход на форму выбора текущей компетенции.

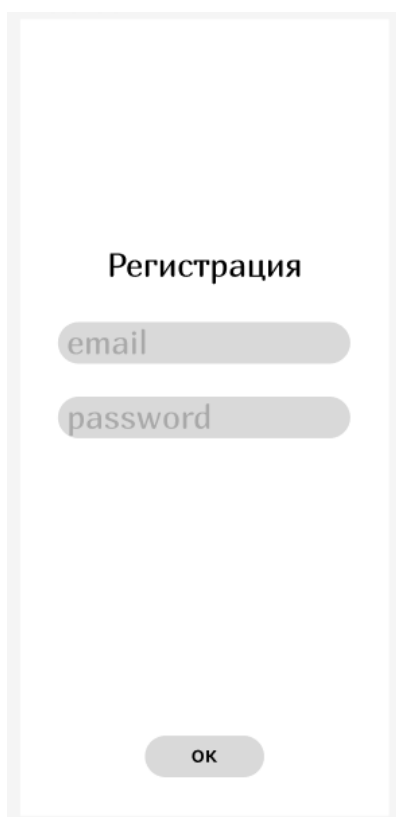
Скриншот экрана регистрации. В центре экрана, выровненном по центру, находится заголовок "Регистрация". Ниже заголовка расположены два поля для ввода: "email" и "password". Оба поля имеют серый фон и закругленные концы. В нижней части экрана, также выровненной по центру, находится кнопка "OK" с тем же серым фоном и закругленными концами. Весь экран имеет белый фон.

Рис. 1. Форма регистрации

- Форма авторизации
 - Нужна для входа пользователя в созданный им ранее аккаунт. Содержит поля ввода электронной почты и пароля для идентификации пользователя и обеспечения доступа к аккаунту. После завершения процесса авторизации происходит переход на форму личного профиля.

The image shows a mobile app screen for login. At the top, the title 'Войти' (Login) is centered. Below it are two input fields: the first is labeled 'email' and the second is labeled 'password'. At the bottom of the screen is a button labeled 'OK'.

Рис. 2. Форма авторизации

- Форма выбора текущей компетенции
 - Нужна для определения основной компетенции только что зарегистрировавшегося пользователя. Содержит в себе два поля с выпадающими списками: специализация и грейд. Специализация представляет собой технологический стек пользователя, грейд – степень продвинутости пользователя в выбранной сфере. После завершения выбора текущей компетенции происходит переход на форму личного профиля.

The image shows a mobile app screen for selecting a competency. The title 'Выбор текущей компетенции' (Select current competency) is centered at the top. Below it are two dropdown menus: the first is labeled 'specialization' and the second is labeled 'grade'. At the bottom of the screen is a button labeled 'OK'.

Рис. 3. Форма выбора текущей компетенции

- Форма главной страницы (личного профиля)
 - Является основной страницей приложения. При наличии аккаунта загружается по умолчанию при запуске приложения. Нужна для отображения списка всех компетенции, как полученных, так и находящихся в процессе получения. Для тех, что находятся в процессе выполнения, можно настроить набор hard и soft скиллов (навыков), перейти к форме обучающих файлов и к форме прохождения контрольного теста. Помимо этого, позволяет добавлять новые грейды за счёт кнопки "+" – она выполняет переход на форму выбора желаемой компетенции. Также имеется дополнительное меню, содержащее два пункта – «Удалить аккаунт» и «Текущие компетенции». Первая кнопка позволяет удалить аккаунт из системы, т.е. полностью стирает все данные об аккаунте из базы данных приложения и выполняет автоматический переход на форму регистрации. Вторая выполняет переход на форму просмотра текущих компетенций соответственно.

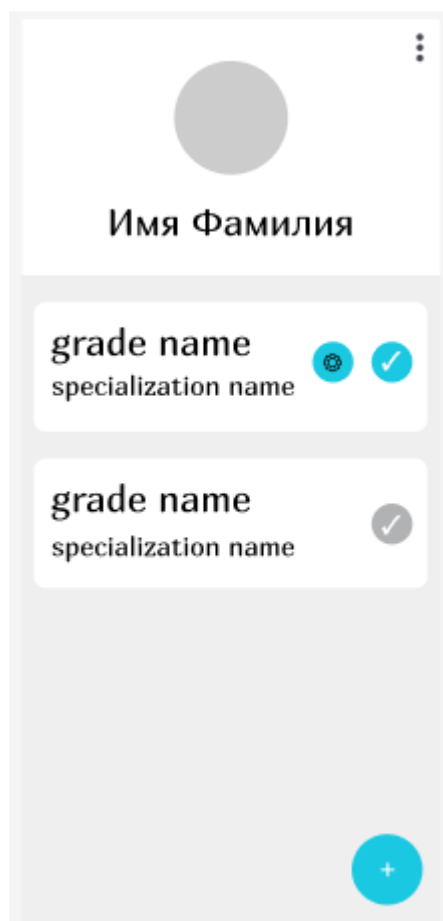
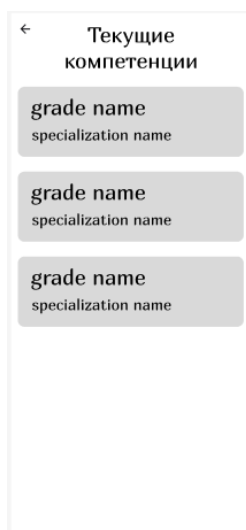


Рис. 4. Форма личного профиля

- Форма просмотра текущих компетенций (полученные компетенции)
 - Нужна для просмотра статистики пользователем – данный экран отображает те компетенции, которые пользователь освоил и в которых успешно завершил выполнение контрольного теста (для

каждой специализации отображается только максимальный грейд из полученных пользователем).



← Текущие компетенции

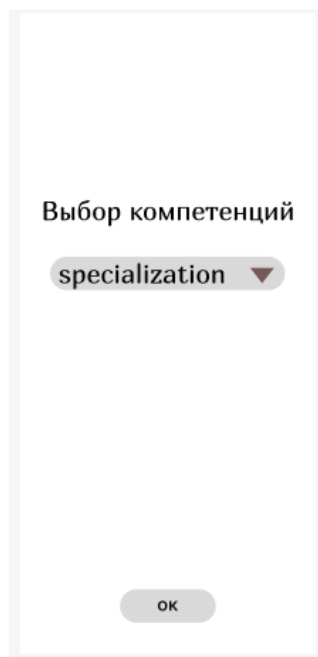
grade name
specialization name

grade name
specialization name

grade name
specialization name

Рис. 5. Форма просмотра текущих компетенций

- Форма выбора желаемой компетенции
 - Нужна для добавления новых компетенций, которые будут находиться в процессе получения пользователем. Содержит поле выпадающего списка специализаций – пользователь выбирает, какую технологию он хочет изучить. Выбор грейда здесь уже недоступен – для новых специализаций по умолчанию будет выбран минимальный грейд, для уже изученных до определенного грейда автоматически выбирается следующая ступень. После завершения процесса выбора компетенции происходит переход на форму выбора набора hard скиллов.



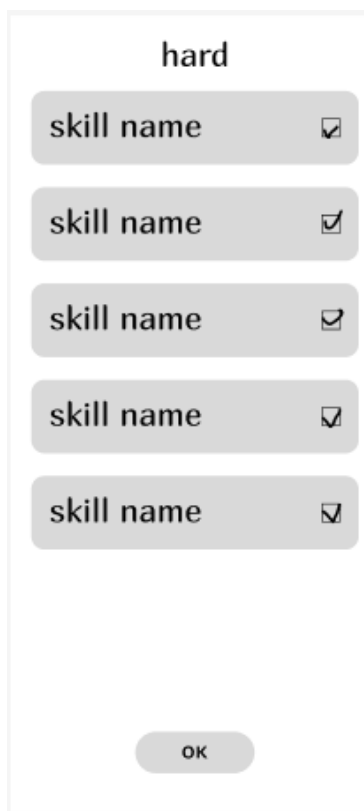
Выбор компетенций

specialization ▼

OK

Рис. 6. Форма выбора желаемой компетенции

- Форма выбора набора hard skills
 - Нужна для определения тех профессиональных скиллов, которыми пользователь уже владеет в совершенстве – по ним обучающие материалы не будут представлены. Обучающие материалы загружаются только по незнакомым для пользователя темам, чтобы не нагружать его лишней информацией. Содержит в себе список скиллов, каждый из которых можно отметить галочкой в чекбоксе, если пользователь считает, что ему не требуется изучение данного скилла. После завершения выбора происходит переход на форму выбора набора soft скиллов.



The image shows a mobile application interface for selecting hard skills. At the top, the word "hard" is centered. Below it, there are five identical rows. Each row consists of a light gray rounded rectangle containing the text "skill name" on the left and a small square checkbox with a checkmark on the right. At the bottom of the form is a light gray rounded rectangle with the text "OK" centered inside it.

Рис. 7. Форма выбора набора hard skills

- Форма выбора набора soft skills
 - Нужна для определения тех личных компетенций, которыми пользователь уже владеет в совершенстве – по ним обучающие материалы не будут представлены. Обучающие материалы загружаются только по незнакомым для пользователя темам, чтобы не нагружать его лишней информацией. Содержит в себе список скиллов, каждый из которых можно отметить галочкой в чекбоксе, если пользователь считает, что ему не требуется изучение данного скилла. После завершения ввода происходит выполнение загрузки файлов для обучения в соответствии с выбранными hard и soft скиллами и затем производится переход на форму просмотра файлов для обучения.

soft

skill name ☒

skill name ☒

skill name ☒

skill name ☒

skill name ☒

OK

Рис. 8. Форма выбора набора soft skills

- Форма просмотра файлов для обучения
 - Нужна для просмотра файлов, загруженных в соответствии с выбранным набором скиллов. Файлы представляют собой обучающие текстовые материалы с определенным форматированием (.md файлы). Помимо этого, в файлах могут содержаться ссылки на дополнительные обучающие ресурсы. Также форма содержит кнопку открытия формы выбора файлов.

≡ filename.md

1. Прототипы экранных форм (не менее 7)
2. По каждой форме:
 - название,
 - краткое описание для чего нужна, какие элементы для чего нужны,
 - картинка
1. Для создания прототипов можно использовать любой онлайн-сервис, например: <https://app.diagrams.net/>, <https://www.figma.com/> и др.
2. Диаграмма сущностей (ER) (не менее 7 сущностей)
 - указать основные атрибуты сущностей
 - Связи М:М реализовать через сущность-связку

Рис. 9. Форма просмотра файлов для обучения

- Форма выбора файлов
 - Нужна для выбора необходимого файла из списка. Файлы группируются по папкам в соответствии с тем скиллом, к изучению которого они относятся. После выбора файла открывается форма просмотра файлов.

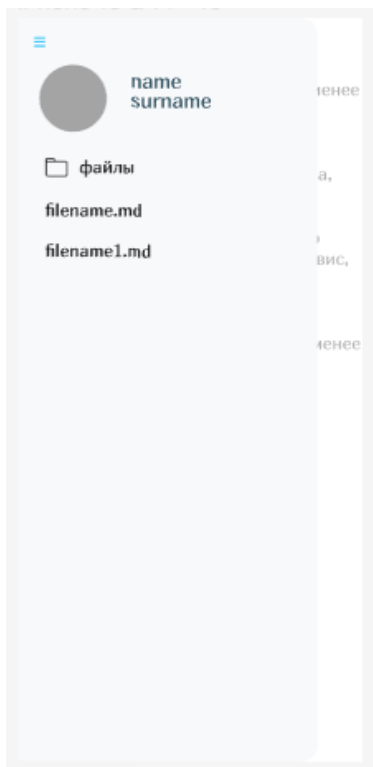


Рис. 10. Форма выбора файлов

- Форма прохождения контрольного теста
 - Нужна для официального завершения процесса получения выбранной компетенции. После того, как пользователь ответит правильно на все вопросы, компетенция автоматически будет считаться полученной, её можно будет найти на экране текущих компетенций; произойдет переход на форму личного профиля. Содержит в себе определённое количество вопросов – настраивается автоматически в зависимости от выбранных скиллов и от их количества. Общая база вопросов по каждой специализации больше, чем количество вопросов в одном тесте – это необходимо для того, чтобы вопросы минимально повторялись от попытки к попытке. Правильные ответы пользователю не отображаются – после завершения теста единственное, что он узнает – все ли было правильно или есть ошибки.

Тест

Вопрос 1

Вопрос 1

Ответ 1

Ответ 1

Ответ 2

Ответ 2

Ответ 3

Вопрос 2

Вопрос 2

Ответ 1

Ответ 1

Ответ 2

Ответ 2

Ответ 3

OK

Рис. 11. Форма прохождения контрольного теста

2. Диаграмма сущностей (ER)

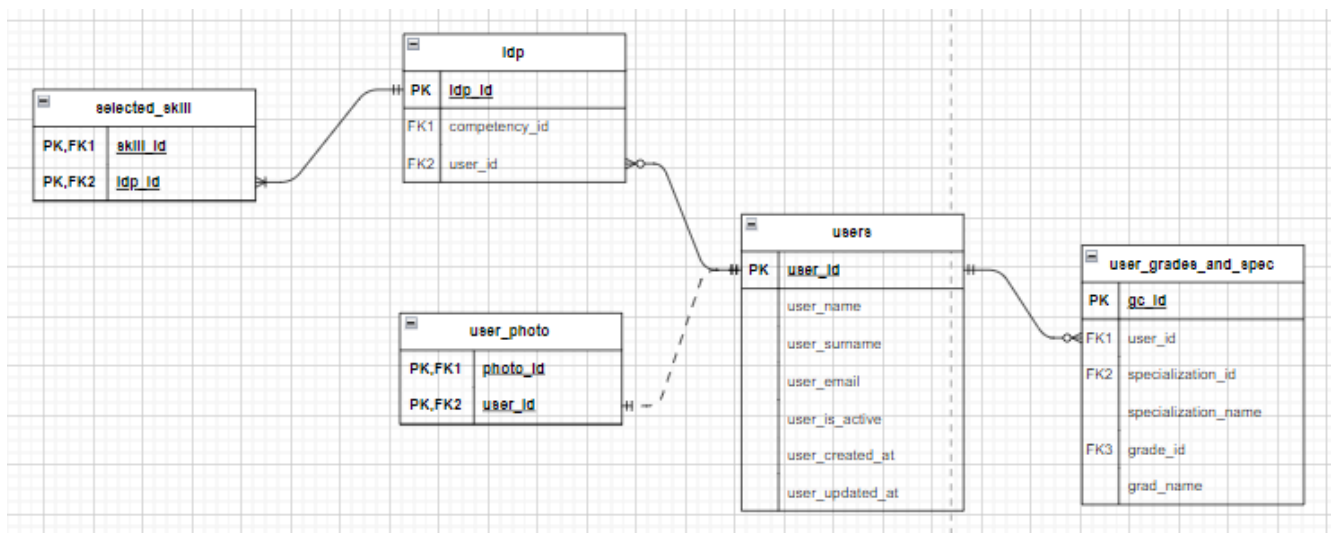


Рис. 12. Сущности, связанные с информацией о пользователе

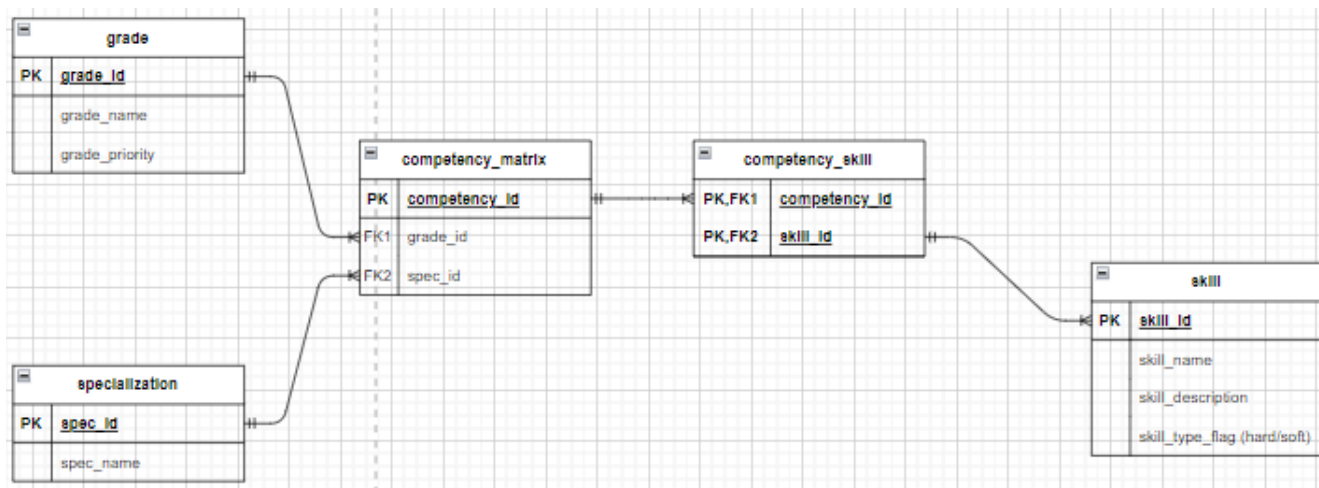


Рис. 13. Сущности, связанные с компетенциями

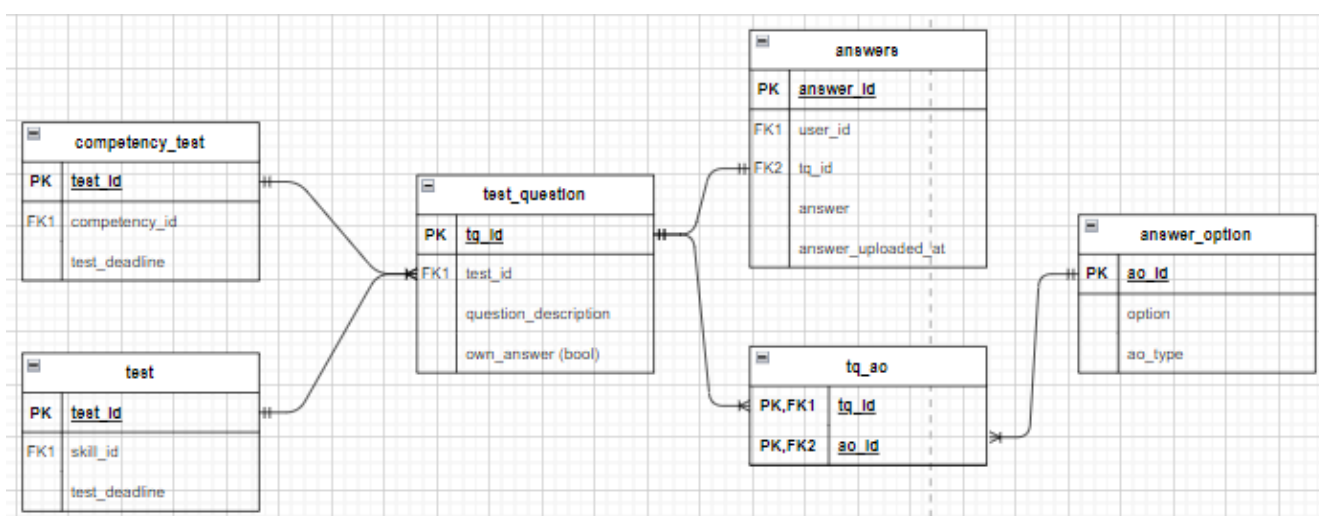


Рис. 14. Сущности, связанные с контрольным тестированием

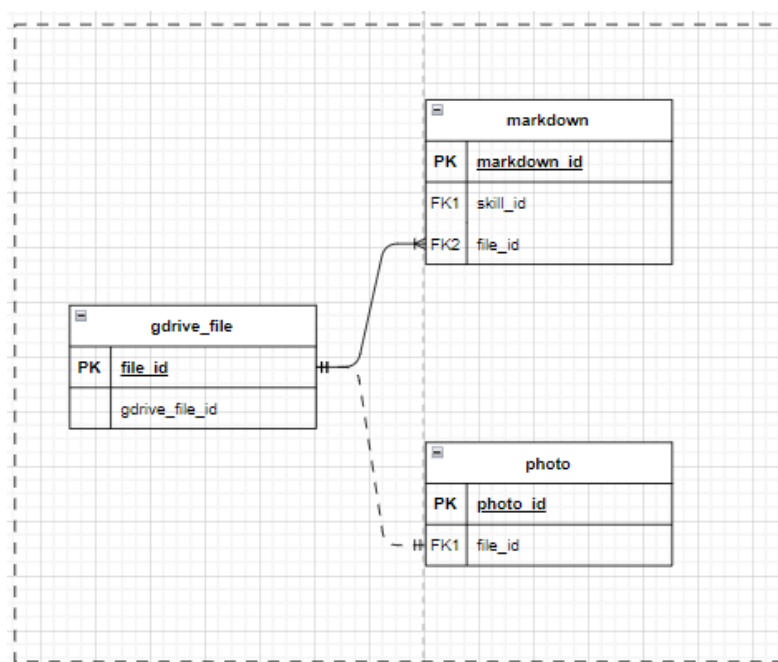


Рис. 15. Сущности, связанные с файлами (маркдаунами)

3. Разработка апи системы

user		
GET	/users	get all users
GET	/users/{id}	get user
PATCH	/users/{id}	patch user

Рис. 16. Функции, связанные с информацией о пользователе

Первый метод отвечает за получение информации по всем пользователям, второй для получения сведений о конкретном аккаунте, третий для изменения данных о пользователе.

Первый метод не требует входных параметров, предназначен для получения информации о всех пользователях, имеющих в базе данных. Выходная информация:

```
[
  {
    "id": 0,
    "name": "string",
    "surname": "string",
    "email": "string",
    "idps": [
      {
        "id": 0,
        "competency_id": 0,
        "selected_skills": [
          {
            "skill_id": 0
          }
        ]
      }
    ]
  },
  "specs_and_grades": [
    {
      "specialization_id": 0,
      "specialization_name": "string",
      "grade_id": 0,
      "grade_name": "string"
    }
  ]
}
```

Parameters

Name	Description
id * required	<input type="text" value="id"/>
integer(\$int64)	(path)
full	Default value : false
boolean	<input type="text" value="false"/>
	(query)

Рис. 17. Входные параметры второго запроса

Responses

Code	Description	Links
200	get user	No links

Media type

Controls Accept header.

Рис. 18. Выходная информация второго запроса

На выходе получим такой же json, как и в первом случае, только для одного пользователя.

PATCH /users/{id} patch user

Parameters Try it out

Name	Description
id * required	<input type="text" value="id"/>
integer(\$int64)	(path)

Request body required

patch user

Example Value | Schema

```
{
  "name": "string",
  "surname": "string",
  "email": "string"
}
```

Рис. 19. Входная информация третьего запроса

В третьем запросе помимо обязательно поля id требуется тело запроса, которое будет содержать новые данные о пользователе. Могут быть изменены имя, фамилия и электронная почта.

idp		
GET	/users/{id}/idps	get user idps
POST	/users/{id}/idps	add user idp
PATCH	/users/{userId}/idps/{idpId}	patch user idp
DELETE	/users/{userId}/idps/{idpId}	delete user idp

Рис. 20. Методы, связанные с индивидуальным планом развития (ИПР) пользователя

Первый метод предназначен для получения информации о всех имеющихся ИПР пользователя. Необходимым входным параметром является только id пользователя. Второй метод добавляет новый ИПР к пользователю. Третий позволяет редактировать текущие ИПР пользователя, четвёртый же удаляет ИПР.

GET	/users/{id}/idps	get user idps
Parameters		
Name	Description	
id * required	<input type="text" value="id"/>	
integer(\$int64)		
(path)		

Рис. 21. Входные данные запроса на получение информации о пользовательских ИПР

Выходная информация первого запроса при успешном его выполнении:

```
{
  "competency_id": 0,
  "selected_skills": [
    {
      "skill_id": 0
    }
  ]
}
```

POST

/users/{id}/idps add user idp

Parameters

Name	Description
id * required	<input type="text" value="id"/>
integer(\$int64)	
(path)	

Request body required

add user idp

Example Value | Schema

```
{  "competency_id": 0,  "selected_skills": [    {      "skill_id": 0    }  ]}
```

Рис. 22. Входная информация второго запроса

В методе, выполняющем добавление ИПР на аккаунт пользователя, помимо id пользователя, необходимым является тело запроса, содержащее id компетенции и id выбранных пользователем скиллов.

Code	Description
200	add user idp

Рис. 23. Выходная информация второго запроса

PATCH

/users/{userId}/idps/{idpId} patch user idp

^

Parameters

Try it out

Name	Description
userId * required integer(\$int64) (path)	<input type="text" value="userId"/>
idpId * required integer(\$int64) (path)	<input type="text" value="idpId"/>
shouldDeleteOrPostSkills * required boolean (query)	Default value : false <input type="text" value="false"/>

Request body required

application/json

patch user idp

Example Value | Schema

```
{  "competency_id": 0,  "selected_skills": [    {      "skill_id": 0    }  ]}
```

Рис. 24. Входная информация третьего запроса

Третий метод, редактирующий имеющийся у пользователя ИПР, требует id пользователя, id ИПР, а также логическую переменную, определяющую добавление или удаление скиллов в данном запросе. Помимо этого, требуется тело запроса с id компетенции и id выбранных пользователем скиллов.

Responses		
Code	Description	Links
200	patch user idp	No links

Рис. 25. Выходная информация третьего запроса

DELETE
/users/{userId}/idps/{idpId}
delete user idp

Parameters
Try it out

Name	Description
userId * required integer(\$int64) (path)	<input type="text" value="userId"/>
idpId * required integer(\$int64) (path)	<input type="text" value="idpId"/>

Responses

Code	Description	Links
200	delete user idp	No links

Рис. 26. Входная и выходная информация четвертого запроса

Для удаления ИПР с аккаунта пользователя требуется указать в запросе id пользователя и id ИПР. Тело запроса не требуется.

specs and grades

GET
/users/{id}/specs-and-grades
get user specs and grades

POST
/users/{id}/specs-and-grades
add user spec and grade

PATCH
/users/{userId}/specs-and-grades/{sagId}
patch user spec and grade

DELETE
/users/{userId}/specs-and-grades/{sagId}
delete user spec and grade

Рис. 27. Методы, связанные с компетенциями и грейдами пользователя

Первый метод выполняет отправку на клиент сведений о специализациях и соответствующих грейдах (уровнях профессионализма) пользователя. Второй позволяет добавить новую специализацию и указать соответствующий грейд. Третий даёт возможность редактировать специализации пользователя. Четвёртый, соответственно, удаляет специализацию с аккаунта.

GET

/users/{id}/specs-and-gradesget user specs and grades

^

Parameters

Try it out

Name	Description
id * required	
integer(\$int64)	id
(path)	

Responses

Code	Description	Links
200	get user specs and grades	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
[
  {
    "specialization_id": 0,
    "specialization_name": "string",
    "grade_id": 0,
    "grade_name": "string"
  }
]
```

Рис. 28. Входная и выходная информация первого запроса

Для выполнения первого запроса требуется только id пользователя. В ответе получаем id специализации, название, а также id грейда и его наименование.

POST

/users/{id}/specs-and-grades add user spec and grade

^

Parameters

Try it out

Name	Description
id * required	id

integer(\$int64)

(path)

Request body required

application/json

add user spec and grade

Example Value | Schema

```
{
  "specialization_id": 0,
  "specialization_name": "string",
  "grade_id": 0,
  "grade_name": "string"
}
```

Responses

Code	Description	Links
200	add user spec and grade	No links

Рис. 29. Входная и выходная информация второго запроса

Для добавления новой компетенции на аккаунт пользователя необходимо передать id нужного пользователя, а также всю ту информацию, что приходит в ответе на первый запрос: id специализации, название, id грейда и его наименование. Всё это указывается в теле запроса.

PATCH

/users/{userId}/specs-and-grades/{sagId} patch user spec and grade

^

Parameters

Try it out

Name	Description
userId * required integer(\$int64) (path)	<input type="text" value="userId"/>
sagId * required integer(\$int64) (path)	<input type="text" value="sagId"/>

Request body required

application/json

patch user spec and grade

Example Value | Schema

```
{
  "specialization_id": 0,
  "specialization_name": "string",
  "grade_id": 0,
  "grade_name": "string"
}
```

Responses

Code	Description	Links
200	patch user spec and grade	No links

Рис. 30. Входная и выходная информация третьего запроса

Данный метод позволяет редактировать информацию об уже добавленной специализации. Необходимо указать id пользователя, id компетенции, а также заполнить тело запроса как для создания новой специализации.

DELETE
/users/{userId}/specs-and-grades/{sagId} delete user spec and grade
^

Parameters
Try it out

Name	Description
userId * required integer(\$int64) (path)	<input type="text" value="userId"/>
sagId * required integer(\$int64) (path)	<input type="text" value="sagId"/>

Responses

Code	Description	Links
200	delete user spec and grade	No links

Рис. 31. Входная и выходная информация при удалении компетенции

При вызове метода удаления требуются только id пользователя и id компетенции.

specialization
^

GET
/specializations specializations
v

grade
^

GET
/grades grades
v

competency
^

GET
/competency competency
v

Рис. 32. Методы, связанные с существующими специализациями, компетенциями и грейдами

Все данные методы предназначены для получения сведений о всех имеющихся специализациях, грейдах и компетенциях соответственно. Ни один из методов не требует входных параметров, ответы возвращаются в виде json объектов.

GET

/specializations specializations

^

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	specializations	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
[
  {
    "spec_id": 0,
    "spec_name": "string"
  }
]
```

Рис. 33. Входная и выходная информация метода получения специализаций

GET

/grades grades

^

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	grades	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
[
  {
    "grade_id": 0,
    "grade_name": "string",
    "grade_priority": 0
  }
]
```

Рис. 34. Входная и выходная информация метода получения грейдов

GET

/competency competency

^

Parameters

Try it out

Name	Description
specId * required integer(\$int64) (query)	<input type="text" value="specId"/>
gradeId * required integer(\$int64) (query)	<input type="text" value="gradeId"/>

Responses

Code	Description	Links
200	<div>competency</div> <div>Media type</div> <div><div>application/json</div></div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "competency_id": 0, "skills": [{ "skill_id": 0, "skill_name": "string", "skill_description": "string", "skill_type_flag": true }] }</pre></div>	

 No links |

Рис. 35. Входная и выходная информация метода получения компетенций

4. Составим иерархическую структуру работ (рис. 36).

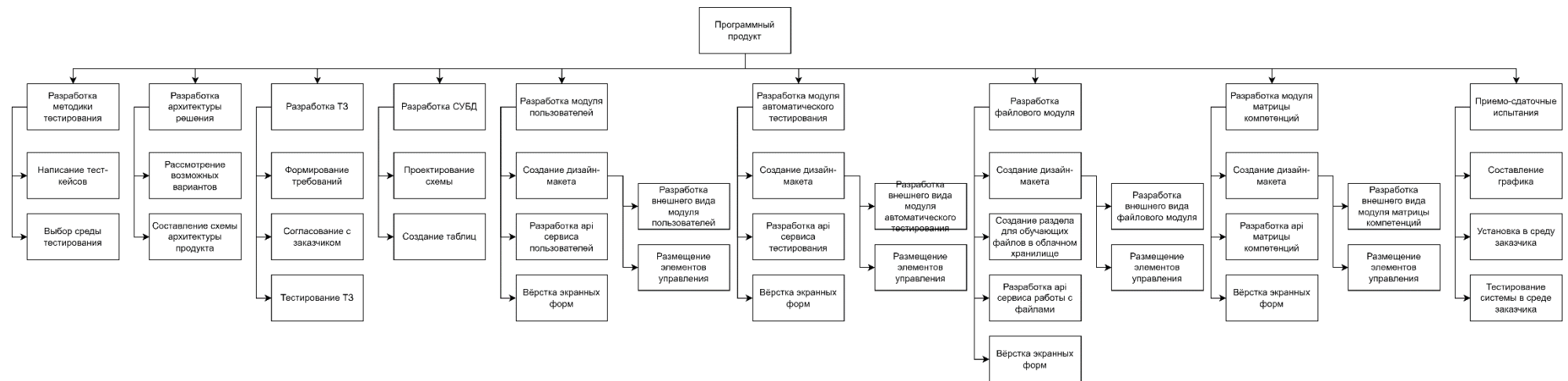


Рис. 36. ИСР

5. Необходимо оценить время выполнения проекта по методу PERT.

Для того, чтобы выполнить данную оценку, необходимо сначала определить количественный состав работ. Имеем на данный момент: 11 экранных форм, 19 сущностей и 14 методов *арі*.

$$N_{\Phi} = 11 \text{ экранных форм}$$

$$O_{\Phi} = 1 \text{ ч - оптимистичная оценка для экранных форм}$$

$$P_{\Phi} = 6 \text{ ч - пессимистичная оценка для экранных форм}$$

$$M_{\Phi} = 2 \text{ ч - средние трудозатраты для экранных форм}$$

$$N_c = 19 \text{ сущностей}$$

$$O_c = 1 \text{ ч - оптимистичная оценка для сущностей}$$

$$P_c = 2 \text{ ч - пессимистичная оценка для сущностей}$$

$$M_c = 1 \text{ ч - средние трудозатраты для сущностей}$$

$$N_m = 14 \text{ методов } \textit{арі}$$

$$O_m = 1 \text{ ч - оптимистичная оценка для } \textit{арі} \text{ методов}$$

$$P_m = 2 \text{ ч - пессимистичная оценка для } \textit{арі} \text{ методов}$$

$$M_m = 1 \text{ ч - средние трудозатраты для } \textit{арі} \text{ методов}$$

$$E_i = (O_i + 4M_i + P_i)/6$$

$$E_{\Phi} = (O_{\Phi} + 4M_{\Phi} + P_{\Phi})/6 = (1 + 4 * 2 + 6)/6 = 2.2$$

$$E_c = (O_c + 4M_c + P_c)/6 = (1 + 4 * 1 + 2)/6 = 1.2$$

$$E_m = (O_m + 4M_m + P_m)/6 = (1 + 4 * 1 + 2)/6 = 1.2$$

$$CKO_i = (P_i - O_i)/6$$

$$CKO_{\Phi} = (P_{\Phi} - O_{\Phi})/6 = (6 - 1)/6 = 0.8$$

$$CKO_c = (P_c - O_c)/6 = (2 - 1)/6 = 0.2$$

$$CKO_m = (P_m - O_m)/6 = (2 - 1)/6 = 0.2$$

$$E = \sum N_i E_i = 11 * 2.2 + 19 * 1.2 + 14 * 1.2 = 64$$

$$CKO = \sqrt{\sum N_i CKO_i^2} = \sqrt{11 * 0.8^2 + 19 * 0.2^2 + 14 * 0.2^2} = 3 \text{ человекочаса}$$

$$E_{\text{общ}} = E + 2CKO = 64 + 3 = 67 \text{ человекочасов}$$

$$E_{\text{итог}} = E_{\text{общ}} * 4 = 268 \text{ человекочасов}$$

$$P_{\text{м}} = \frac{E_{\text{итог}}}{132} = 2 \text{ месяца} - \text{будет делать один человек}$$

6. Базовое расписание в виде диаграммы Ганта. В базовом расписании должны быть отражены все элементы ИСР.

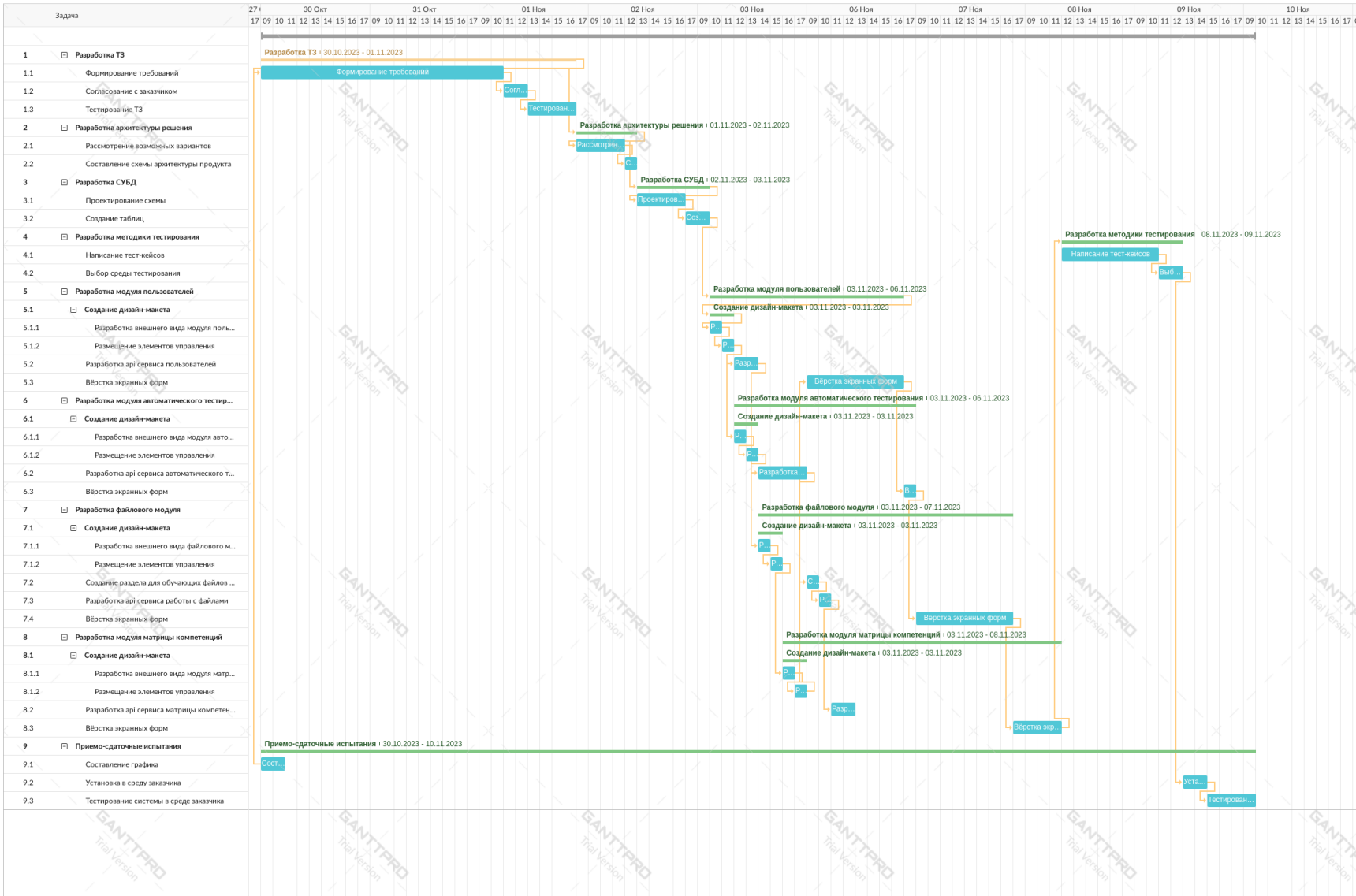


Рис. 37. Диаграмма Ганта