

**TOMSK POLYTECHNIC UNIVERSITY**  **ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**Министерство науки и высшего образования РФ**

Федеральное государственное автономное образовательное учреждение

высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

---

Инженерная школа информационных технологий и робототехники

Направление подготовки 09.04.02 «Информационные системы и технологии»

Отделение информационных технологий

Методические указания к лабораторной работе №8

**TPU GITLAB**

по дисциплине

**Управление проектами разработки программного обеспечения**

## 1. ЦЕЛЬ РАБОТЫ

Целью лабораторной работы является приобретение навыков работы с *Git*.

*Git* – это распределённая система управления версиями, созданная Линусом Торвальдсом для управления разработкой ядра *Linux*.

## 2. ЗАДАНИЕ

Изучить основы работы с *Git*. Создать модель сбора требований от двух заказчиков, используя платформу <https://gitlab.tpu.ru>.

В качестве основы взять свой вариант для лабораторных работ [в папке преподавателя на ftp-сайте кафедры ВТ ТПУ](#).

### 2.1. Алгоритм выполнения задания

#### 1. Авторизоваться на сайте [gitlab.tpu.ru](https://gitlab.tpu.ru)

Чтобы авторизоваться на сайте, нужно перейти по ссылке [https://gitlab.tpu.ru/users/sign\\_in](https://gitlab.tpu.ru/users/sign_in), затем выбрать авторизацию в домене ТПУ, ввести корпоративные логин и пароль и нажать кнопку *Sign in*, как показано на рисунке 1.



## Tomsk Polytechnic University

### GitLab Community Edition

По вопросам технической поддержки обращаться в Главный информационный узел.

E-mail: [admin@tpu.ru](mailto:admin@tpu.ru)

Телефон: +7 (3822) 70-16-16

TPU	Standard
TPU Username	
aas189	
Password	
.....	
<input checked="" type="checkbox"/> Remember me	
Sign in	

Рисунок 1 – Авторизация на [gitlab.tpu.ru](https://gitlab.tpu.ru)

## 2. Создать новый проект

Для создания проекта нужно нажать кнопку *Create a project* на главной странице (рисунок 2).

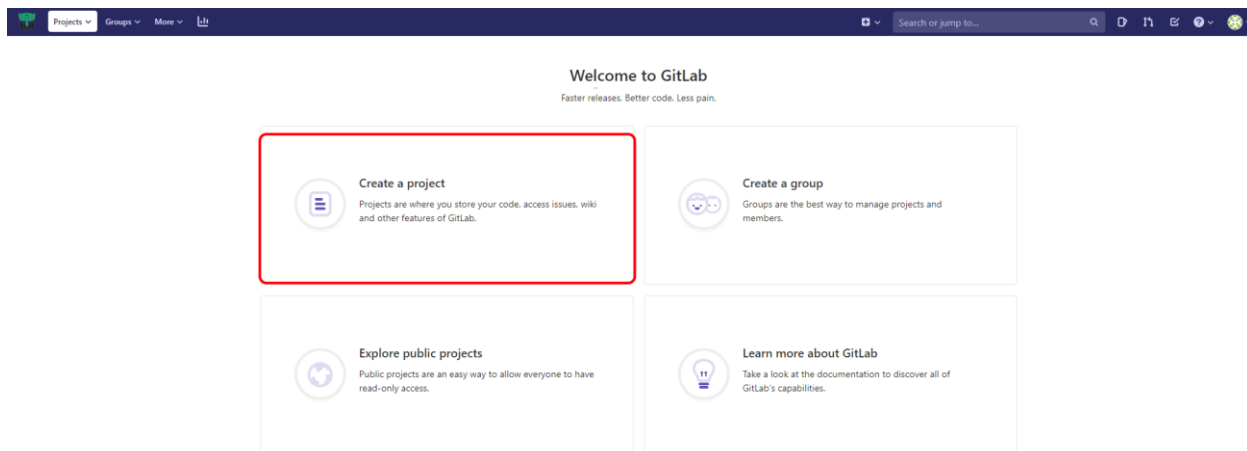


Рисунок 2 – Главная страница репозитория

На открывшейся странице заполнить поля с названием и описанием проекта и нажать кнопку *Create project*. Название и описание проекта можно взять из разделов 1.1 и 1.2 устава проекта (лабораторная работа №1) соответственно. Пример создания проекта приведен на рисунке 3.

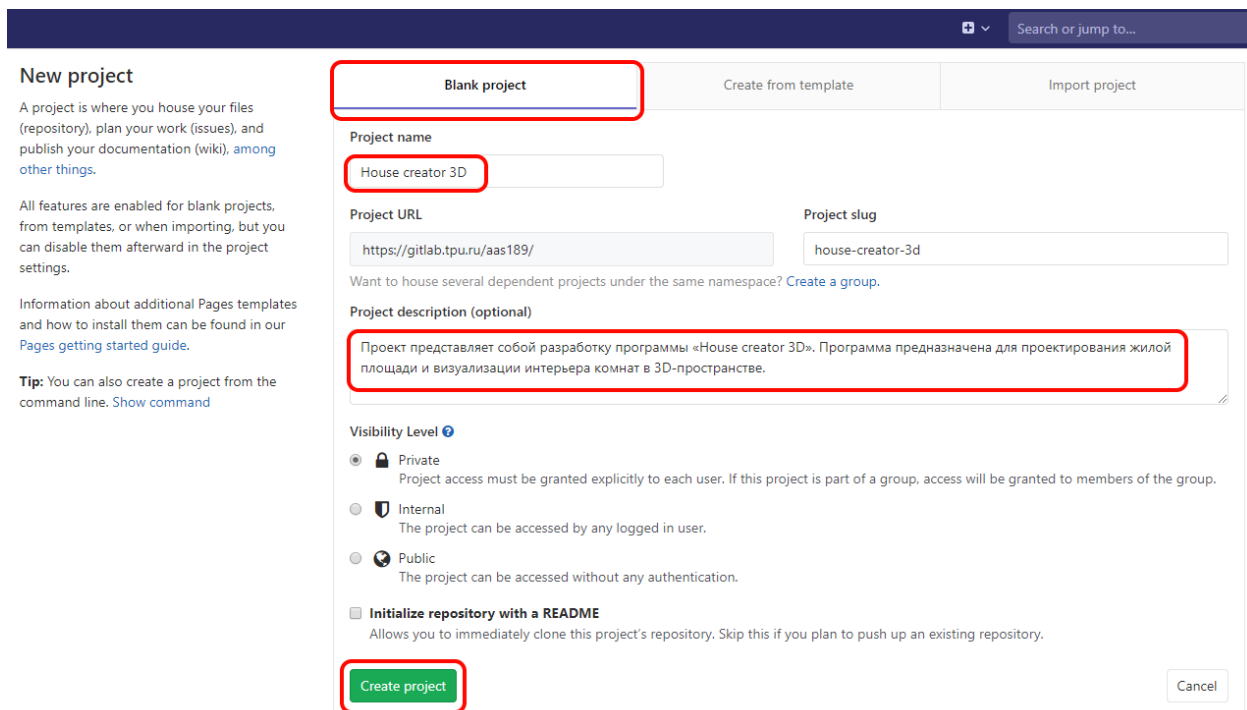


Рисунок 3 – Создание проекта

### 3. Создать новый файл

После создания проекта произойдет переход на страницу проекта. Теперь можно добавить в него новый файл. Для этого нужно нажать кнопку *New file*, как показано на рисунке 4. После чего нужно ввести название файла *Requirements* и нажать кнопку *Commit changes* (рисунок 5). Через пользовательский интерфейс *GitLab* название файла нужно вводить на английском языке, иначе возникнет [ошибка при попытке разрешения конфликтов объединения ветвей](#)).

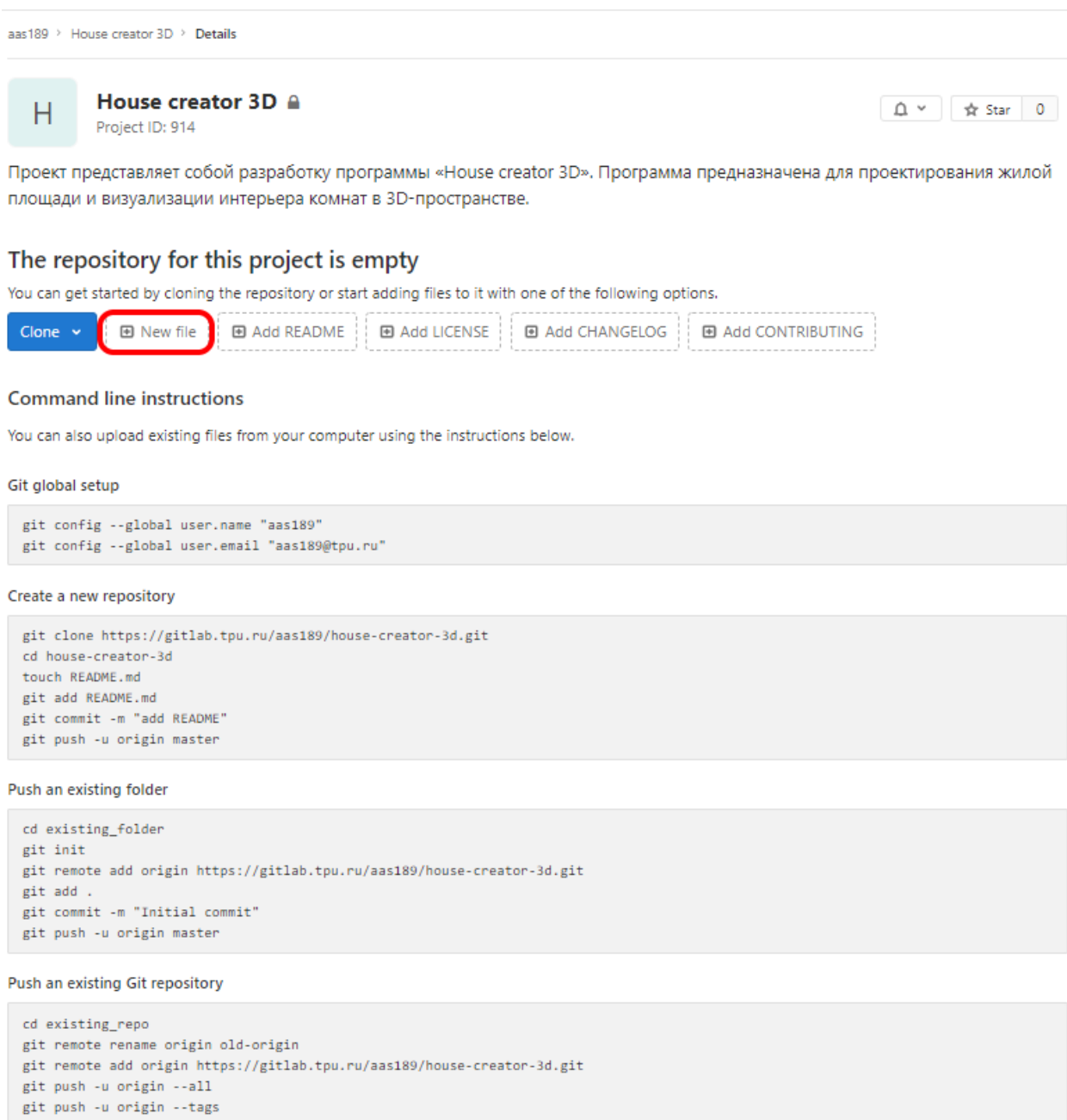


Рисунок 4 – Страница проекта

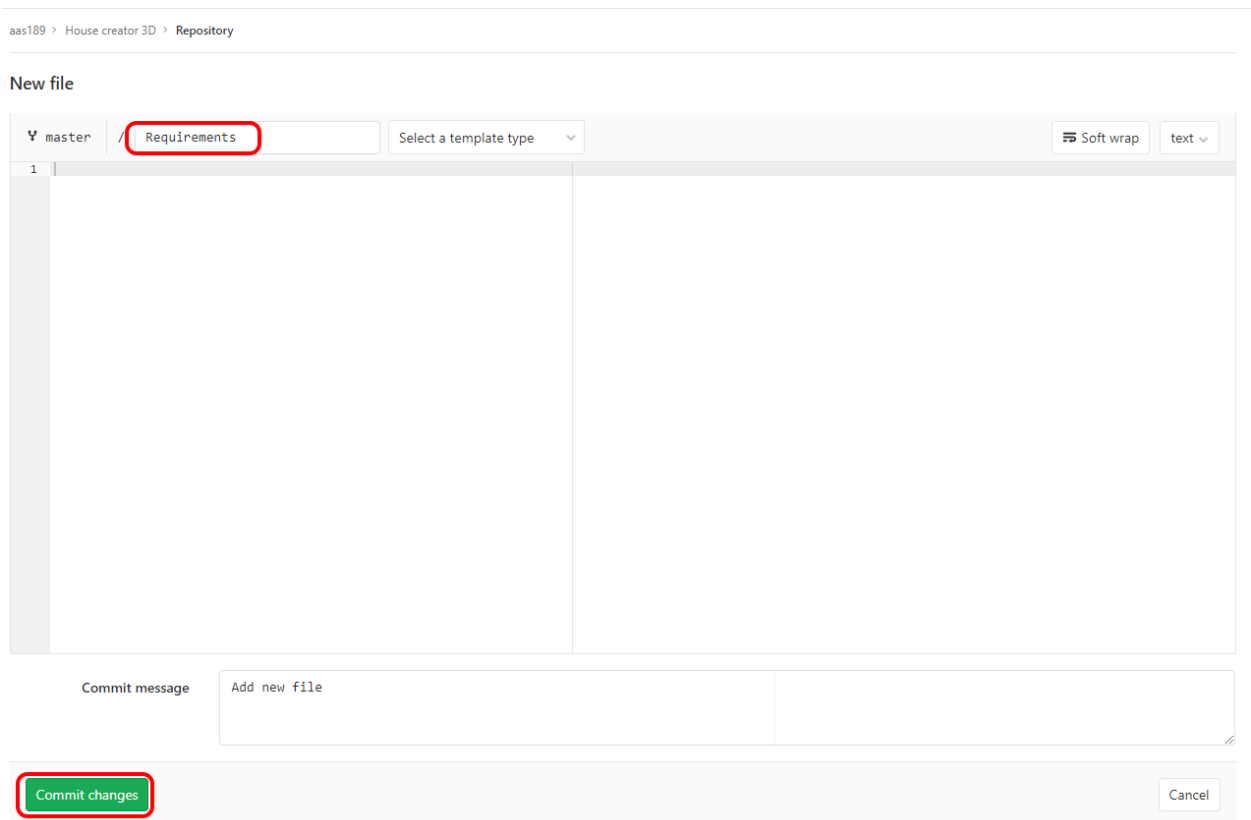


Рисунок 5 – Создание файла

#### 4. Создать новую ветку и отредактировать файл в ней

Теперь создадим новую ветку. Для этого в навигационной панели слева выберем пункт *Branches*, на открывшейся странице нажмем кнопку *New branch* (рисунок 6).

Далее в появившейся форме введем название новой ветки, например, *develop*, и нажмем кнопку *Create branch* (рисунок 7).

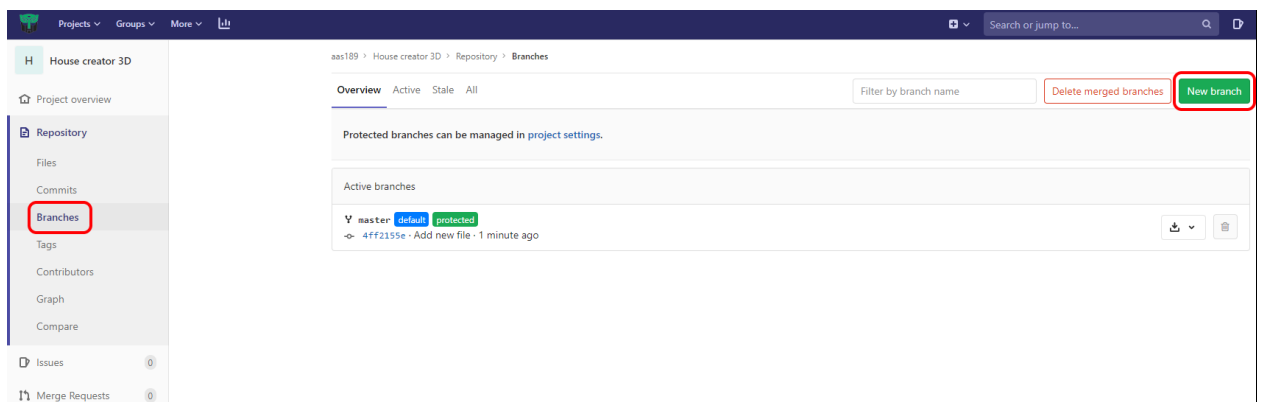


Рисунок 6 – Список ветвей проекта

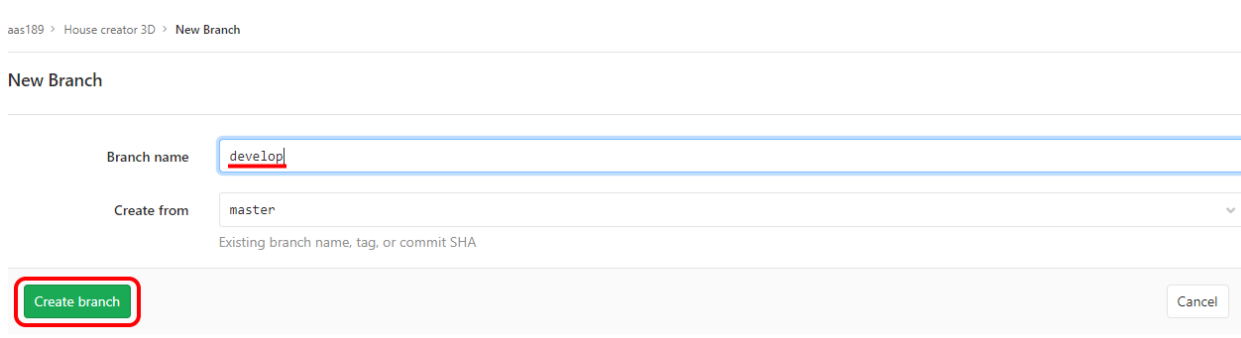


Рисунок 7 – Создание новой ветви

Теперь нужно отредактировать файл с потребностями в новой ветке. Для этого нужно кликнуть по названию файла (рисунок 8), после чего произойдет переход на страницу с информацией о файле. Здесь нужно нажать кнопку *Edit* (рисунок 9) и заполнить файл требованиями из технического задания своего варианта или из пункта 2.2 устава проекта (лабораторная работа №1). Пример заполнения файла представлен на рисунке 10.

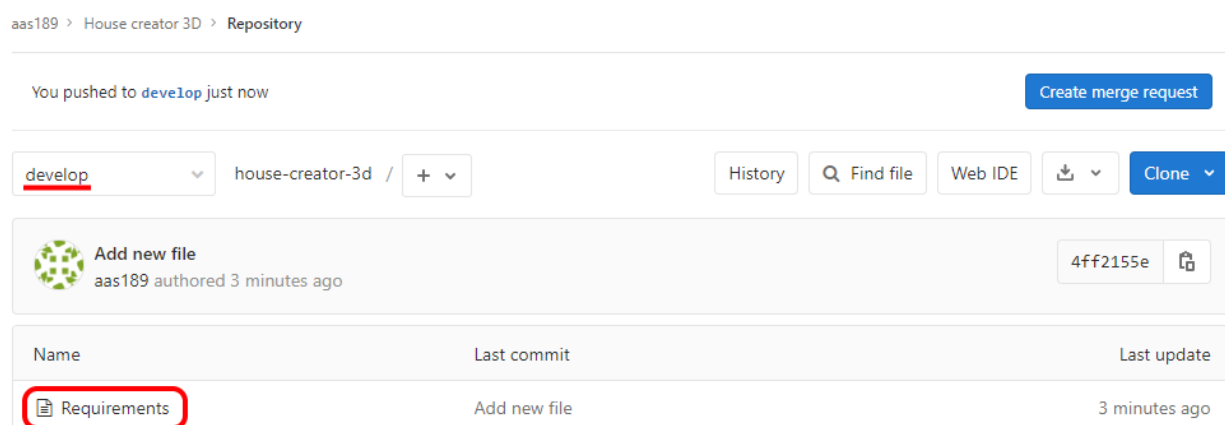


Рисунок 8 – Содержимое репозитория

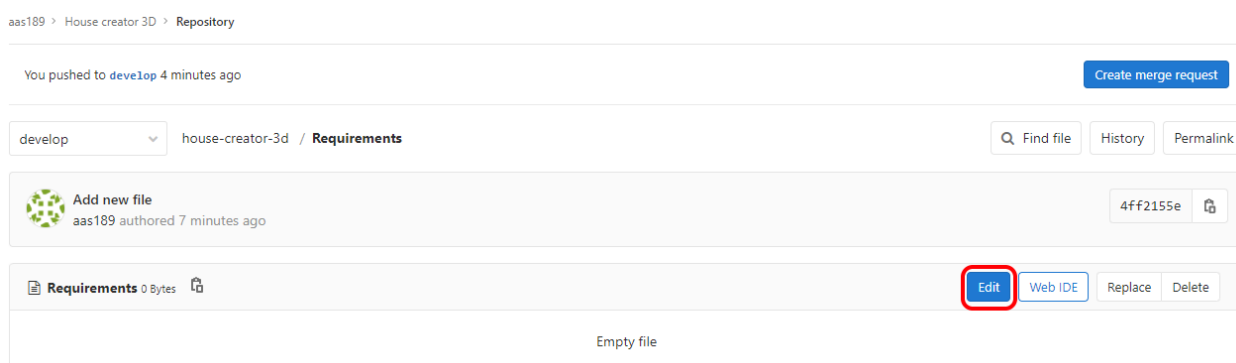


Рисунок 9 – Информация о файле

## Edit file

Write Preview changes

Y develop Requirements Soft wrap text

1. Программа должна предоставлять возможность создавать и редактировать макет
2. Программа должна предоставлять возможность сохранения незаконченного макета в файл и загрузки программы из него
3. Управление должно осуществляться с помощью мыши
4. Программа должна предупреждать о невозможности перемещения предметов вне рабочего поля

Commit message Update Requirements

Target Branch develop

Commit changes Cancel

## Рисунок 10 – Редактирование файла

Снова нажмем кнопку *Edit* и отредактируем файл еще раз, изменив старые требования и/или добавив новые требования, например, как на рисунке 11.

## Edit file

Write Preview changes

Y develop Requirements Soft wrap text

1. Программа должна предоставлять возможность создавать и редактировать макет
2. Программа должна предоставлять возможность сохранения незаконченного макета в файл и загрузки программы из него
3. Управление должно осуществляться с помощью мыши
4. Программа должна предупреждать о невозможности перемещения предметов вне рабочего поля
5. Программа должна автоматически сохранять макет каждые 10 минут

Commit message Update Requirements

Target Branch develop

Commit changes Cancel

## Рисунок 11 – Повторное редактирование файла

## 5. Отредактировать файл в исходной ветке

Теперь нужно отредактировать файл с потребностями в исходной ветке. Для этого перейдем в ветку *master*, как показано на рисунке 12, и нажмем кнопку *Edit*.

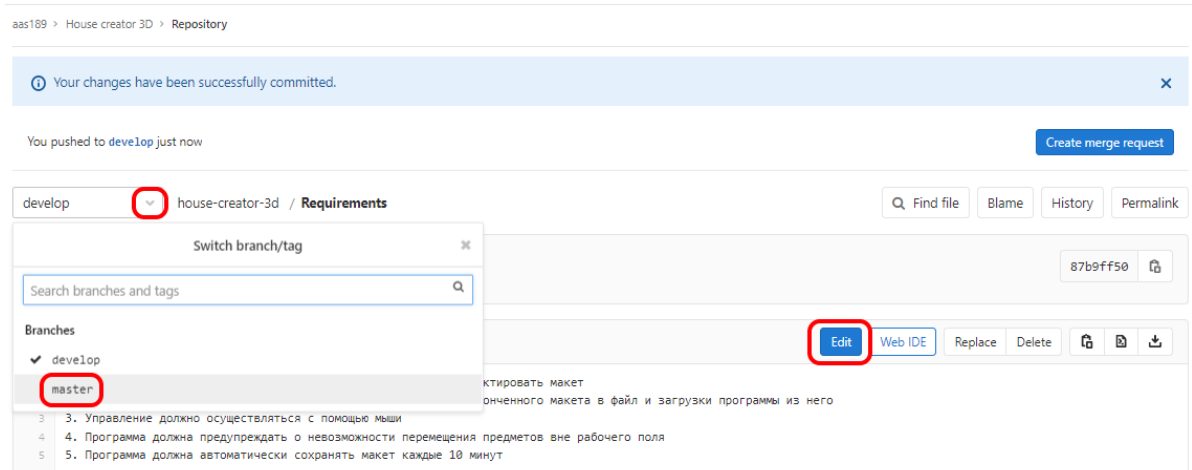


Рисунок 12 – Переход на другую ветку

В открывшейся форме введем те же требования, что в другой ветке, немного изменив их, например, как на рисунке 13. Это будет имитацией требований другого заказчика. После ввода всех требований нажмем кнопку *Commit changes*.

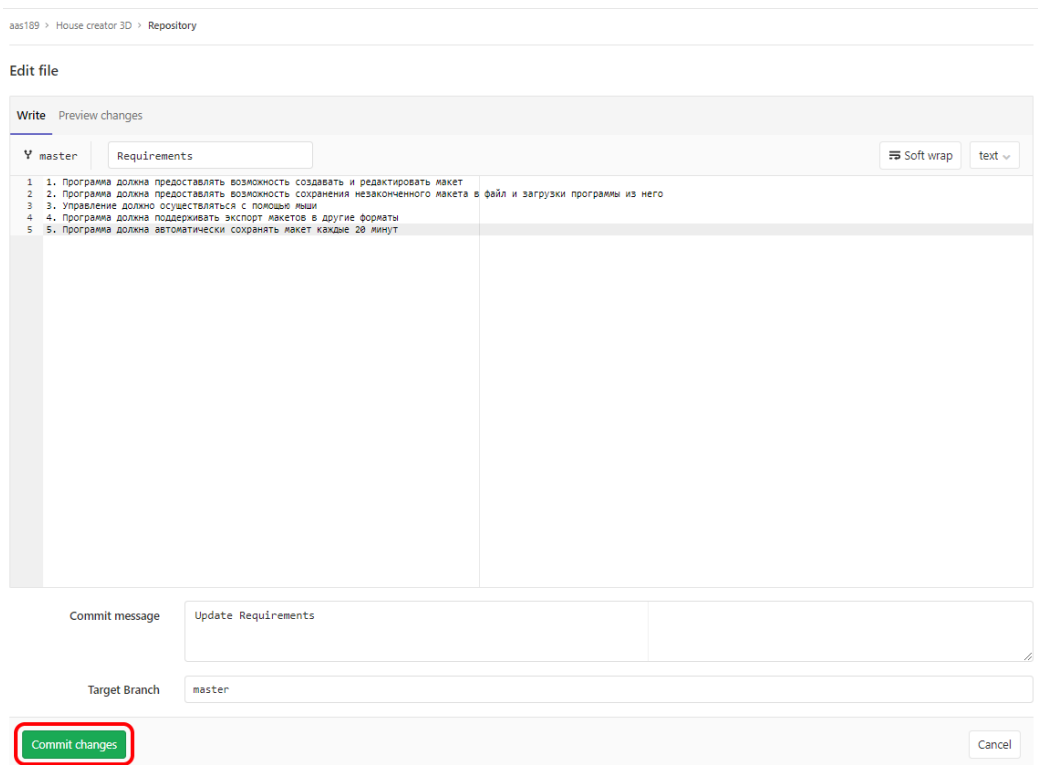


Рисунок 13 – Редактирование файла в другой ветке



## 6. Посмотреть граф изменений в репозитории

Посмотрим граф изменений в репозитории проекта, для этого выберем пункт *Repository – Graph* в панели навигации слева. После внесения изменений, описанных выше, граф репозитория должен выглядеть так же, как на рисунке 14, то есть он должен состоять из двух независимых веток, в каждой из которых находится файл с потребностями.

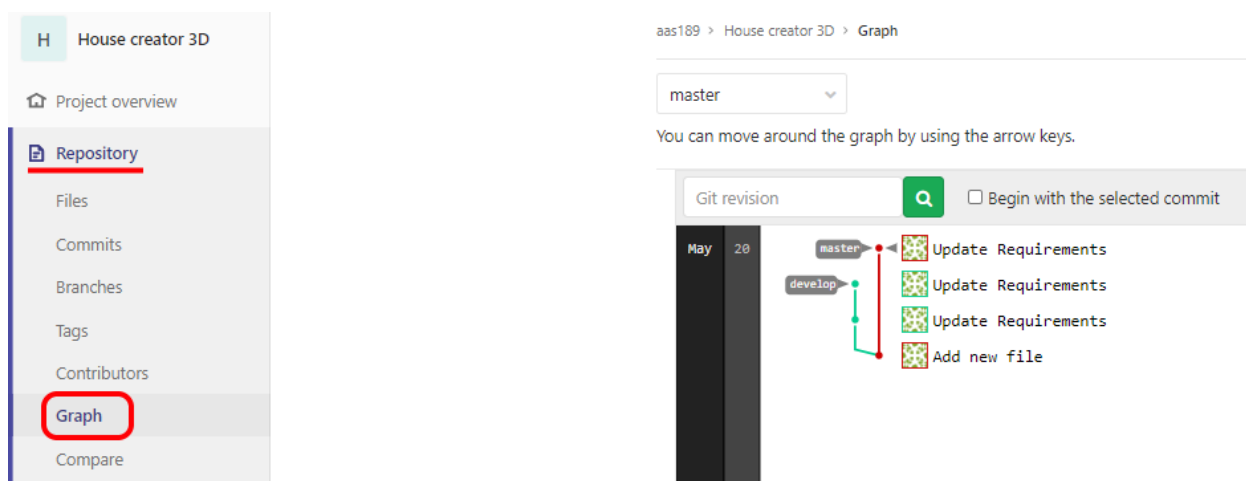


Рисунок 14 – Граф репозитория

## 7. Создать запрос на объединение веток

Теперь создадим запрос на объединение веток. Для этого последовательно:

1. Выберем пункт *Merge Requests* в панели навигации слева и нажмем кнопку *New merge request* (рисунок 15);

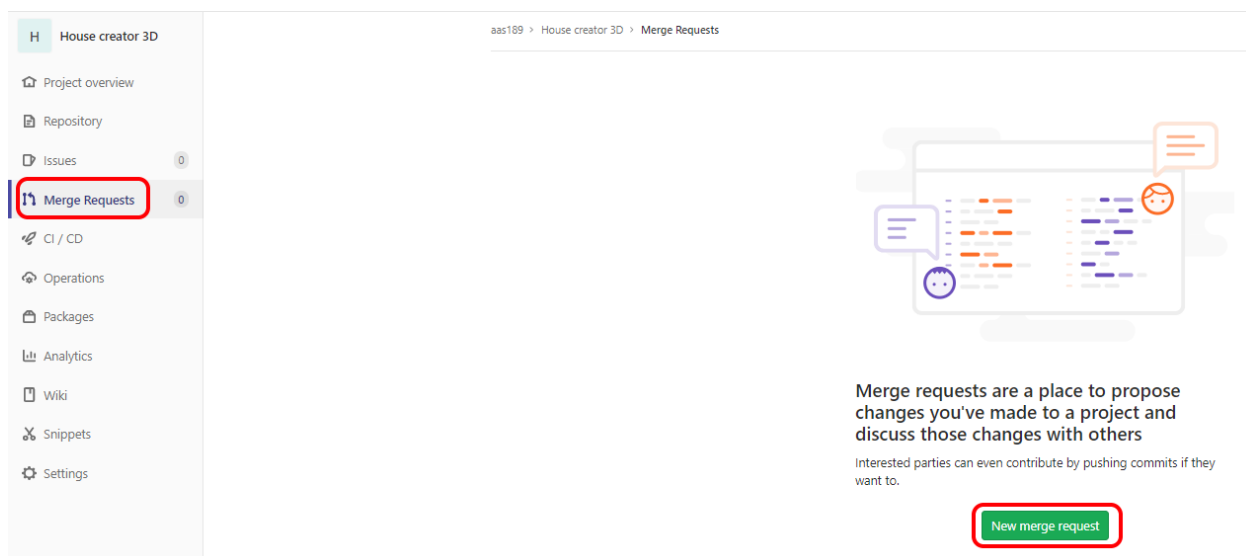


Рисунок 15 – Запросы на слияние ветвей проекта

2. Выберем в качестве ветки-источника ветку *develop*, в качестве ветки назначения – ветку *master* и нажмем кнопку *Compare branches and continue* (рисунок 16);

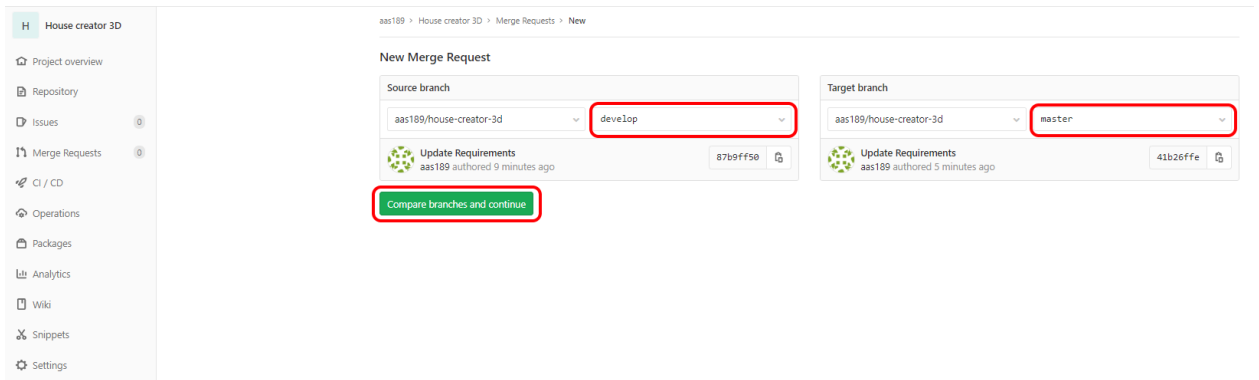


Рисунок 16 – Выбор ветвей для объединения

3. Отредактируем название запроса на объединение (по желанию), уберем галочку с пункта *Delete source branch when merge request is accepted* и нажмем кнопку *Submit merge request* (рисунок 17).

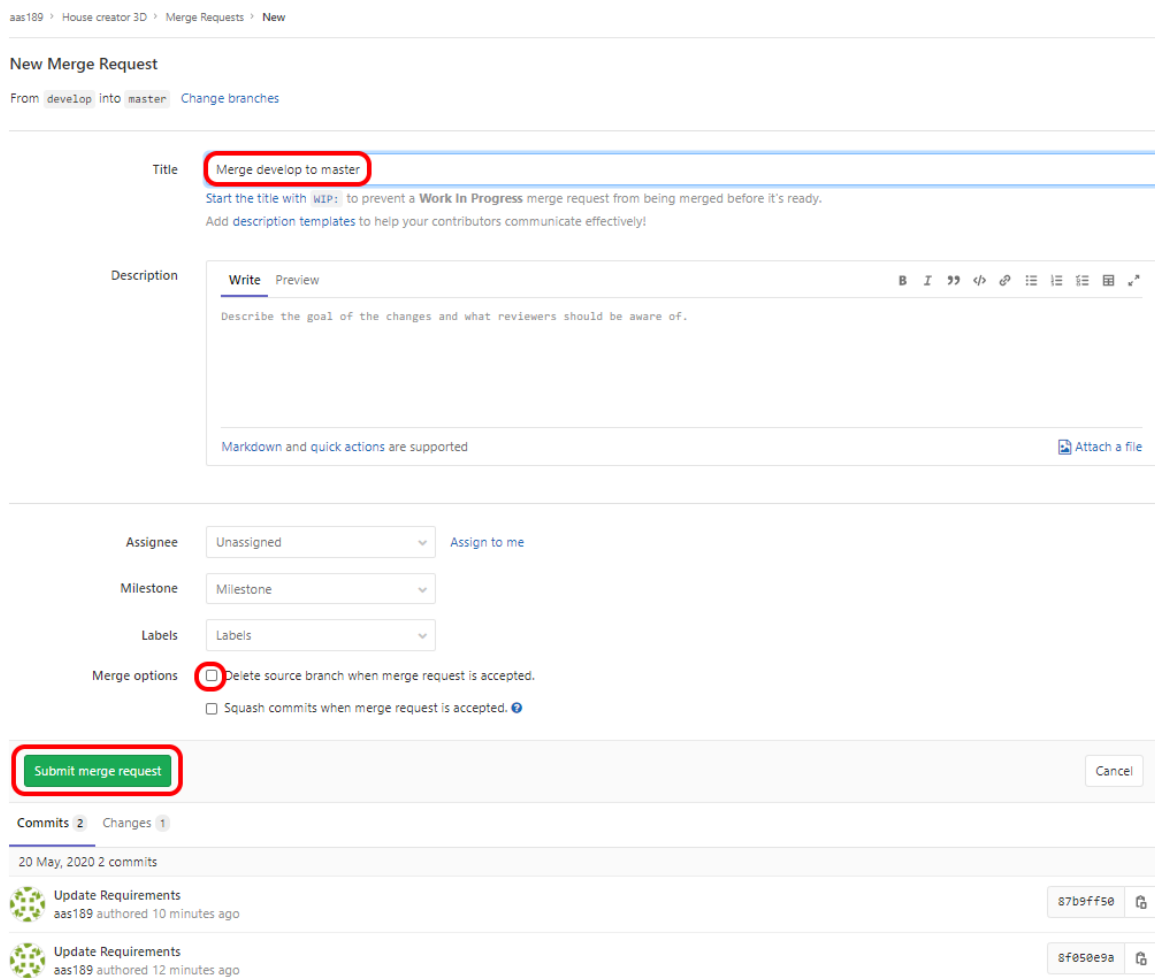
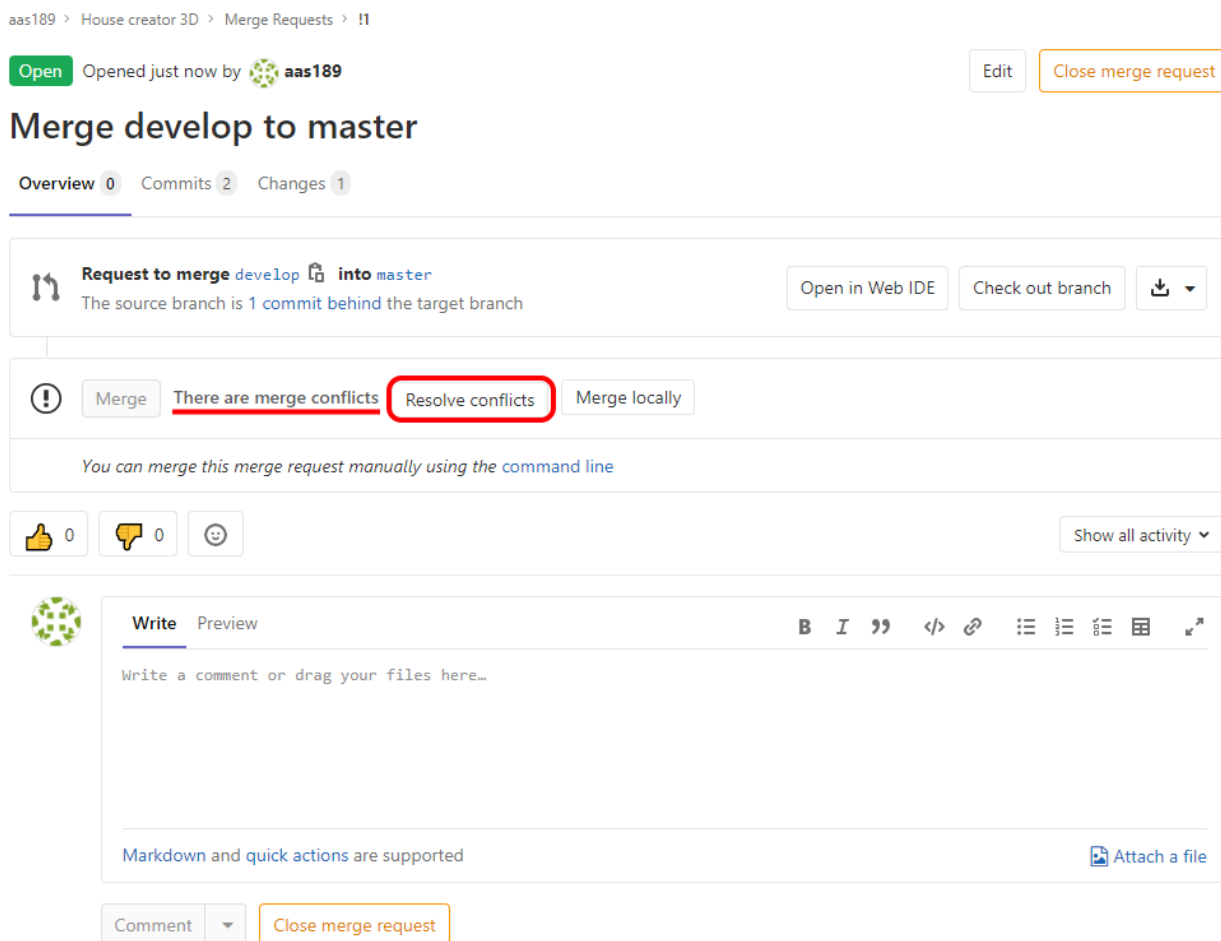


Рисунок 17 – Создание запроса на объединение ветвей

## 8. Разрешить конфликты слияния

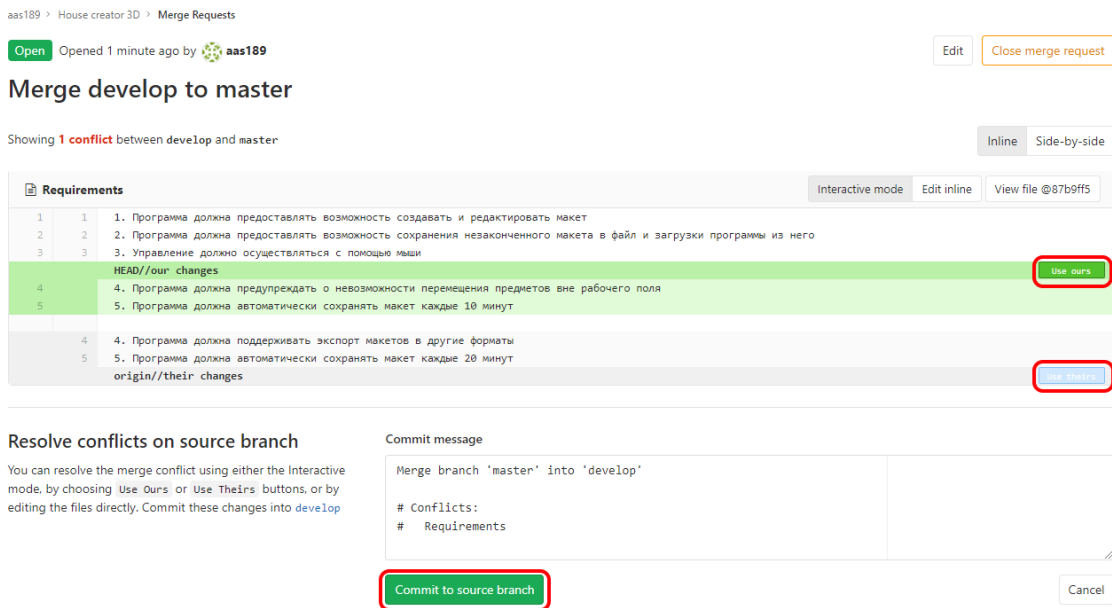
Как видно из рисунка 18, при попытке объединения ветвей возникли конфликты. Это связано с тем, что изменения в них касаются одного и того же файла, но при этом не совпадают. Нужно разрешить конфликты, нажав кнопку *Resolve conflicts*.



The screenshot shows a GitHub Merge Request for merging 'develop' into 'master'. The interface includes a header with the user 'aas189', an 'Open' button, and a 'Close merge request' button. Below the title 'Merge develop to master', there are tabs for 'Overview', 'Commits', and 'Changes'. A section titled 'Request to merge develop into master' indicates that the source branch is 1 commit behind the target branch. A prominent message states 'There are merge conflicts' with a red underline, and a 'Resolve conflicts' button is highlighted with a red box. Other buttons include 'Merge locally' and 'Merge'. Below this, there are thumbs up/down icons and a 'Show all activity' dropdown. At the bottom, there is a 'Write' comment section with a rich text editor and an 'Attach a file' button.

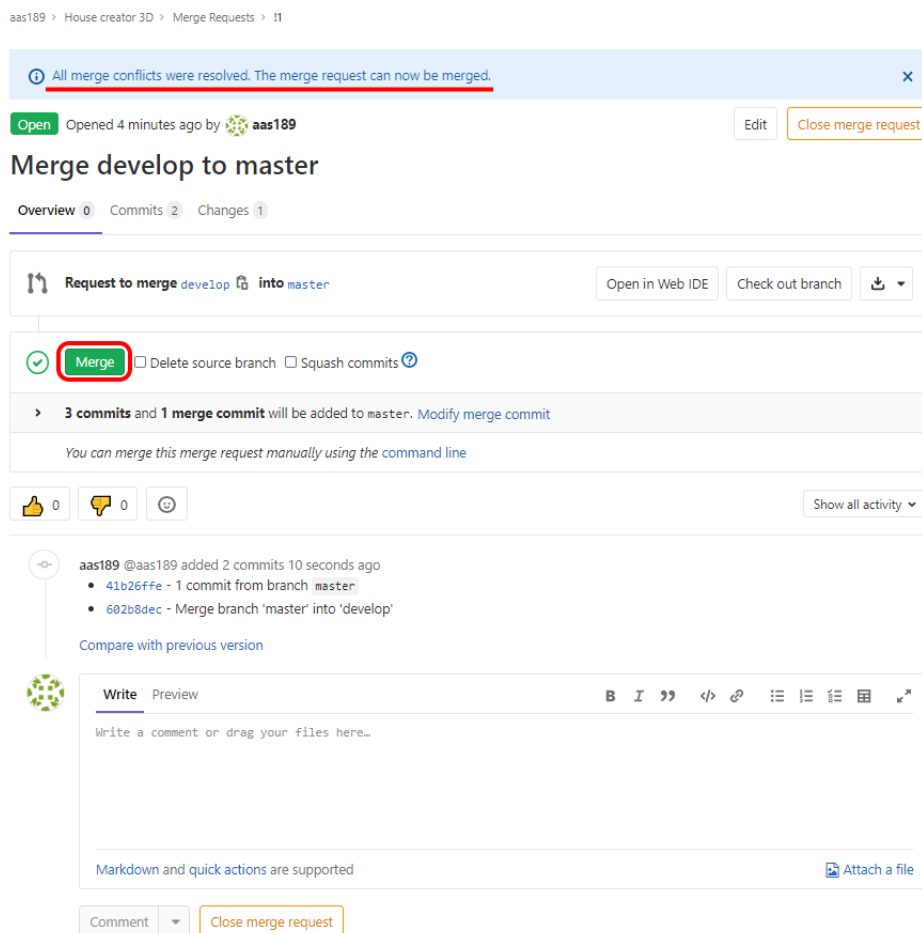
Рисунок 18 – Конфликты при объединении ветвей

Произойдет переход на страницу разрешения конфликтов (рисунок 19). Здесь можно выбрать вариант, находящийся в одной из веток (*Use ours* или *Use theirs*), нажав на соответствующую кнопку, или отредактировать файл вручную, после чего нужно нажать кнопку *Commit to source branch*.



## Рисунок 19 – Разрешение конфликтов при объединении ветвей

После того, как конфликты будут разрешены, появится соответствующее сообщение и кнопка *Merge*. Для завершения объединения ветвей нужно нажать кнопку *Merge* (рисунок 20).



## Рисунок 20 – Объединение ветвей

После объединения ветвей статус запроса изменится на *Merged* (рисунок 21).

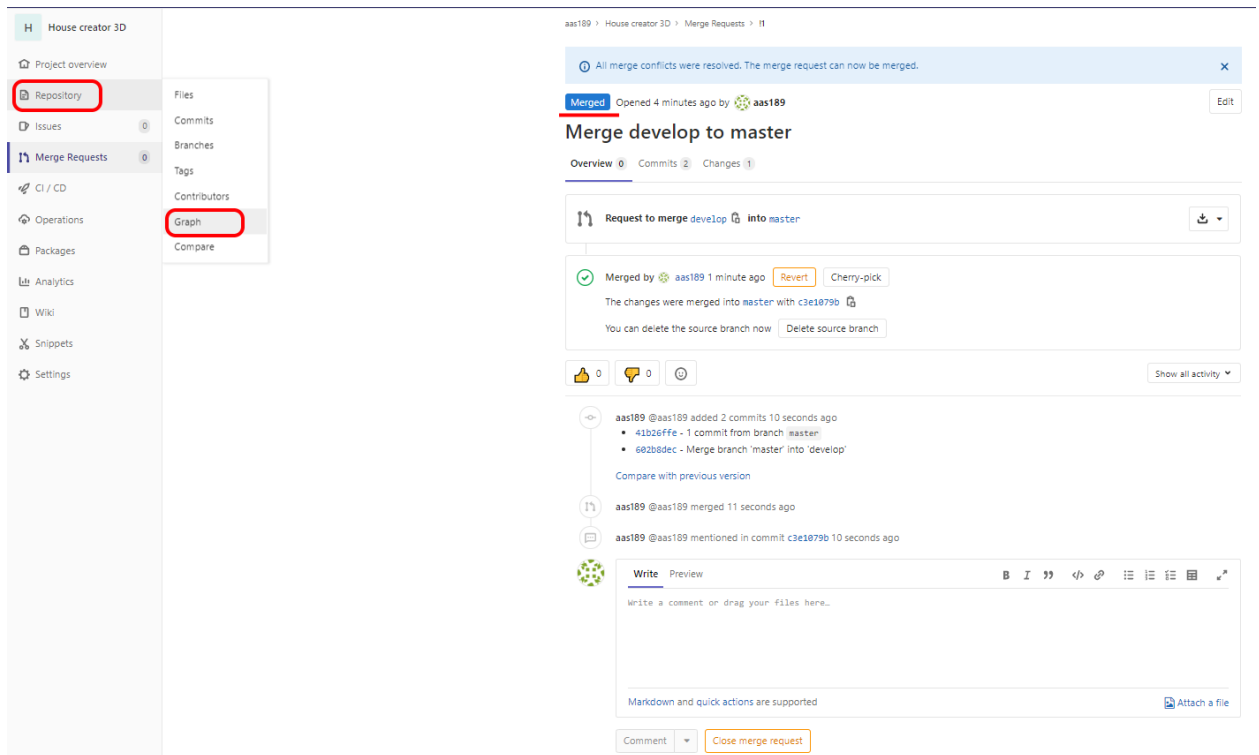


Рисунок 21 – Выполненный запрос на объединение ветвей

## 9. Посмотреть граф изменений в репозитории

Теперь снова посмотрим на граф репозитория, перейдя в пункт *Repository – Graph* (рисунок 21) в панели навигации слева. После слияния ветвей граф репозитория должен выглядеть так же, как на рисунке 22.

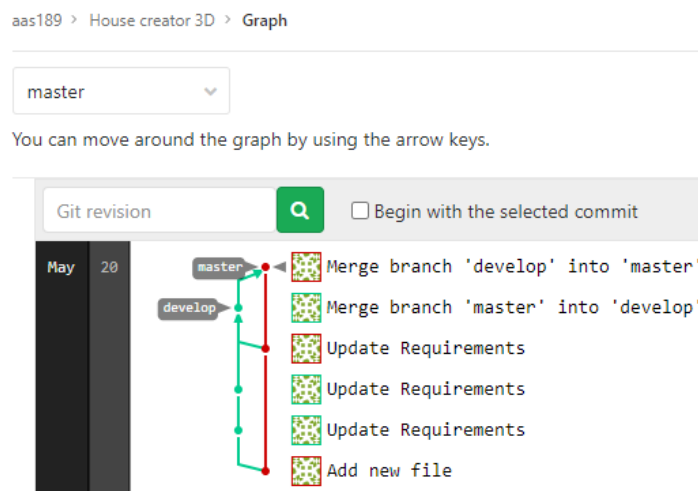


Рисунок 22 – Граф репозитория после объединения ветвей

## 2.2. Пример выполнения задания

Результатом работы является отчет по произведенным действиям, содержащий скриншоты, иллюстрирующие ход выполнения задания. Пример отчета приведен ниже.

### Ход работы

#### 1. Авторизация на gitlab.tpu.ru



## Tomsk Polytechnic University

### GitLab Community Edition

По вопросам технической поддержки обращаться в Главный информационный узел.

E-mail: [admin@tpu.ru](mailto:admin@tpu.ru)

Телефон: +7 (3822) 70-16-16

TPU	Standard
TPU Username	
<input type="text" value="aas189"/>	
Password	
<input type="password" value="....."/>	
<input checked="" type="checkbox"/> Remember me	
<input type="button" value="Sign in"/>	

Рисунок 23 – Авторизация на gitlab.tpu.ru

#### 2. Создание проекта

##### New project

A project is where you house your files (repository), plan your work (issues), and publish your documentation (wiki), among other things.

All features are enabled for blank projects, from templates, or when importing, but you can disable them afterward in the project settings.

Information about additional Pages templates and how to install them can be found in our [Pages getting started guide](#).

**Tip:** You can also create a project from the command line. [Show command](#)

Blank project	Create from template	Import project
Project name		
<input type="text" value="House creator 3D"/>		
Project URL		Project slug
<input type="text" value="https://gitlab.tpu.ru/aas189/"/>		<input type="text" value="house-creator-3d"/>
Want to house several dependent projects under the same namespace? <a href="#">Create a group</a> .		
Project description (optional)		
<input type="text" value="Проект представляет собой разработку программы «House creator 3D». Программа предназначена для проектирования жилой площади и визуализации интерьера комнат в 3D-пространстве"/>		
Visibility Level		
<input checked="" type="radio"/> Private Project access must be granted explicitly to each user. If this project is part of a group, access will be granted to members of the group.		
<input type="radio"/> Internal The project can be accessed by any logged in user.		
<input type="radio"/> Public The project can be accessed without any authentication.		
<input type="checkbox"/> Initialize repository with a README Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.		
<input type="button" value="Create project"/>		<input type="button" value="Cancel"/>

Рисунок 24 – Создание проекта

### 3. Создание файла

aas189 > House creator 3D > Repository

New file

master / Requirements Select a template type Soft wrap text

1

Commit message Add new file

Commit changes Cancel

Рисунок 25 – Создание файла

### 4. Создание ветки и редактирование файла в ней

aas189 > House creator 3D > New Branch

New Branch

Branch name develop

Create from master Existing branch name, tag, or commit SHA

Create branch Cancel

Рисунок 26 – Создание новой ветки

### Edit file

Write Preview changes

develop Requirements Soft wrap text

```
1 1. Программа должна предоставлять возможность создавать и редактировать макет
2 2. Программа должна предоставлять возможность сохранения незаконченного макета в файл и загрузки программы из него
3 3. Управление должно осуществляться с помощью мыши
4 4. Программа должна предупреждать о невозможности перемещения предметов вне рабочего поля
```

Commit message Update Requirements

Target Branch develop

Commit changes Cancel

Рисунок 27 – Редактирование файла в новой ветке

### Edit file

Write Preview changes

develop Requirements Soft wrap text

```
1 1. Программа должна предоставлять возможность создавать и редактировать макет
2 2. Программа должна предоставлять возможность сохранения незаконченного макета в файл и загрузки программы из него
3 3. Управление должно осуществляться с помощью мыши
4 4. Программа должна предупреждать о невозможности перемещения предметов вне рабочего поля
5 5. Программа должна автоматически сохранять макет каждые 10 минут
```

Commit message Update Requirements

Target Branch develop

Commit changes Cancel

Рисунок 28 – Повторное редактирование файла в новой ветке



## 5. Редактирование файла в исходной ветке

aas189 > House creator 3D > Repository

Edit file

Write Preview changes

master Requirements Soft wrap text

1 1. Программа должна предоставлять возможность создавать и редактировать макет  
2 2. Программа должна предоставлять возможность сохранения незаконченного макета в файл и загрузки программы из него  
3 3. Управление должно осуществляться с помощью мыши  
4 4. Программа должна поддерживать экспорт макетов в другие форматы  
5 5. Программа должна автоматически сохранять макет каждые 20 минут

Commit message Update Requirements

Target Branch master

Commit changes Cancel

Рисунок 29 – Редактирование файла в исходной ветке

## 6. Граф изменений в репозитории

aas189 > House creator 3D > Graph

master

You can move around the graph by using the arrow keys.

Git revision   Begin with the selected commit

May 28

Update Requirements  
Update Requirements  
Update Requirements  
Add new file

Рисунок 30 – Граф репозитория

## 7. Создание запроса на объединение веток

aas189 > House creator 3D > Merge Requests > New

### New Merge Request

From `develop` into `master` [Change branches](#)

Title

Start the title with `WIP:` to prevent a **Work In Progress** merge request from being merged before it's ready.  
Add [description templates](#) to help your contributors communicate effectively!

Description

[Write](#) [Preview](#)

**B** *I* `</>`     

Describe the goal of the changes and what reviewers should be aware of.

Markdown and quick actions are supported

[Attach a file](#)

Assignee  [Assign to me](#)

Milestone

Labels

Merge options  Delete source branch when merge request is accepted.  
 Squash commits when merge request is accepted. [?](#)

[Submit merge request](#)

[Cancel](#)

[Commits](#) 2 [Changes](#) 1

20 May, 2020 2 commits



Update Requirements  
aas189 authored 10 minutes ago

87b9ff50



Update Requirements  
aas189 authored 12 minutes ago

8f050e9a



Рисунок 31 – Создание запроса на объединение ветвей

## 8. Разрешение конфликтов слияния

aas189 > House creator 3D > Merge Requests > !1

**Open** Opened just now by aas189

Edit

Close merge request

### Merge develop to master

Overview 0 Commits 2 Changes 1

**Request to merge develop into master**  
The source branch is 1 commit behind the target branch

Open in Web IDE Check out branch

Merge There are merge conflicts Resolve conflicts Merge locally

You can merge this merge request manually using the command line

0 0 Show all activity ▾

Write Preview **B I ” </> ↻**

Write a comment or drag your files here...

Markdown and quick actions are supported Attach a file

Comment ▾ Close merge request

Рисунок 32 – Конфликты при объединении ветвей

aas189 > House creator 3D > Merge Requests

**Open** Opened 1 minute ago by aas189

Edit

Close merge request

### Merge develop to master

Showing 1 conflict between develop and master

Inline

Side-by-side

**Requirements** Interactive mode Edit inline View file @87b9ff5

1	1	1. Программа должна предоставлять возможность создавать и редактировать макет	
2	2	2. Программа должна предоставлять возможность сохранения незаконченного макета в файл и загрузки программы из него	
3	3	3. Управление должно осуществляться с помощью мыши	
<b>HEAD/our changes</b>			
4		4. Программа должна предупреждать о невозможности перемещения предметов вне рабочего поля	
5		5. Программа должна автоматически сохранять макет каждые 10 минут	
<b>origin/their changes</b>			
4		4. Программа должна поддерживать экспорт макетов в другие форматы	
5		5. Программа должна автоматически сохранять макет каждые 20 минут	

#### Resolve conflicts on source branch

You can resolve the merge conflict using either the Interactive mode, by choosing Use Ours or Use Theirs buttons, or by editing the files directly. Commit these changes into develop

#### Commit message

Merge branch 'master' into 'develop'

# Conflicts:  
# Requirements

Commit to source branch

Cancel

Рисунок 33 – Разрешение конфликтов при объединении ветвей

## 9. Граф изменений в репозитории

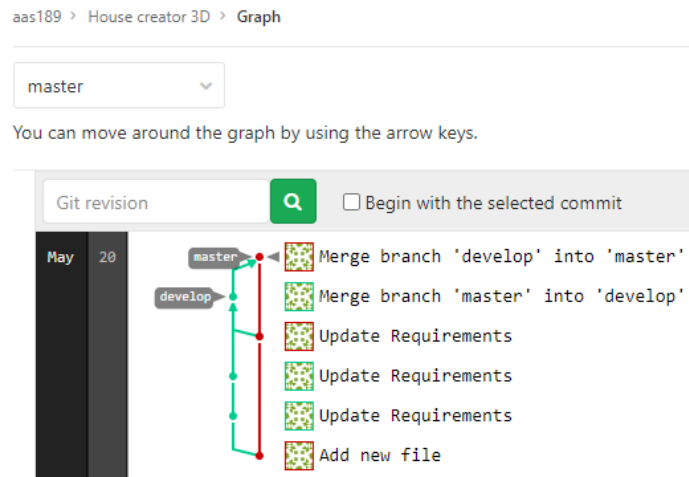


Рисунок 34 – Граф репозитория после объединения ветвей